7-1-2011

Flapping wing design for a dragonfly-like micro air vehicle

Daniel Prosser

Flapping Wing Design for a Dragonfly-Like Micro Air Vehicle

by

Daniel T. Prosser

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Mechanical Engineering

Advised by

Dr. Agamemnon Crassidis, Assistant Professor, Mechanical Engineering
Department of Mechanical Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York
July, 2011

Approved by:

Dr. Agamemnon Crassidis, Associate Professor *Thesis Advisor, Department of Mechanical Engineering*

Dr. Amitabha Ghosh, Professor *Committee Member, Department of Mechanical Engineering*

Dr. Jason Kolodziej, Assistant Professor *Committee Member, Department of Mechanical Engineering*

Dr. Edward Hensel, Professor *Department Representative, Department of Mechanical Engineering*

Thesis Release Permission Form

Rochester Institute of Technology
Kate Gleason College of Engineering

Title:

Flapping Wing Design for a Dragonfly-Like Micro Air Vehicle

I, Daniel T. Prosser, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

Daniel T. Prosser

Date

Acknowledgments

I would like to thank my advisor, Dr. Crassidis, for his help in guiding me during my studies and in allowing me to become involved in this interesting research, for the Air Force in providing the contractual funding under project #FA9550-10-C-0036, and for the assistance of my peers and colleagues also working on this project at Impact Technologies, LLC and Georgia Institute of Technology. I am also greatly thankful for the thoughts, prayers, and motivation given me by my parents, family, friends, and girlfriend. Above all, I thank God for blessing me with the intelligence and determination to finish this task; may all be done for his glory.

Abstract

In this thesis, the aerodynamics of the Quad-Wing Vehicle, a Micro Air Vehicle designed to hover with four flapping wings in a dragonfly-like configuration, is investigated using Computational Fluid Dynamics (CFD), potential flows analysis, and experimental testing. The CFD analysis investigates the kinematics-parameters design space and identifies values of kinematics parameters that maximize the vertical force production in hovering mode while minimizing the aerodynamic power requirement. It also investigates other important considerations, such as the effect of scaling, multi-wing interactions, and comparison with other flapping configurations. In the potential flows analysis, an unsteady 2D panel code is developed and compared with CFD for a broad range of hovering-flight simulations. The results show that, with further development, panel codes may be useful to designers of hovering flapping MAVs because of their time-saving potential compared to CFD. The experimental testing focuses on isolating the aerodynamic forces from other measured forces on a benchtop flapping device, and the findings of the experimental study will be useful for later researchers using experimental methods to study flapping MAV aerodynamics.

Contents

List of Figures

List of Tables

Nomenclature

A	Amplitude of horizontal oscillation; coefficient matrix
a	Chord of the wing at the root
α_0	Airfoil angle of attack halfway through a stroke
B	Total rotation angle for airfoil; circulation strength vector
b	Single-wing span
C	Force coefficient
c	Chord direction or chord length
d	Characteristic length
δ	Stroke plane inclination from vertical
f	Flapping frequency, Hz
F_c	Inertial force due to centripetal accleration
F_h	Usable vertical force in hover
$\vec{F_i}$	Inertial force due to angular acceleration
F	Force
Γ	Total circulation
$\gamma(s)$	Bound circulation distribution
$\underline{\gamma}$	Wake circulation
h	Non-dimensional amplitude
$\mathbf{I}$	Inertia tensor
$_j$	Vortex panel counter
k	Reduced frequency
$_l$	Wake vortex counter

$\vec{M}$	Aerodynamic moment
MAV	Micro Air Vehicle
$\vec{M}_i$	Inertial moment due to angular acceleration
n	Direction normal to the wing; number of airfoil panels
$\hat{n}$	Direction normal to the airfoil surface
ω	Flapping frequency or motor speed, rad/s
$\vec{\Omega}$	Flapping rotational velocity vector
P	Power to overcome aerodynamic moments
p	Static pressure
$_{pan}$	Influence in panel coordinates
Φ	Velocity potential
ϕ	Flapping angle: angle of the wing above the horizontal X-Y plane
$\ddot{\phi}$	Flapping angular acceleration
$\dot{\phi}$	Flapping rotational speed
ϕ_{max}	Maximum flapping amplitude
Π	Non-dimensional group
p_∞	Upstream pressure
$_p$	Current time step
p_y	Wing pivot y-position, benchtop actuator
p_z	Wing pivot z-position, benchtop actuator
QV	Quad-Wing Vehicle
r	Distance to an arbitrary point P or crank radius
$\vec{r}$	Displacement vector
$\vec{r_a}$	Vector from body-axis origin to point on airfoil surface
r_c	Distance from center of rotation to a point on the wing
ρ	Fluid density
$\bar{\rho}$	Mass per unit wing area

S	Wing area
s	Span direction or airfoil surface ordinate
t	Time
θ	Pitching (feathering) angle of the wing about its leading edge
θ_{max}	Maximum feathering amplitude
T	Flapping period
$\bar{U}$	Average oscillation velocity
$\vec{v}$	Velocity of a point on the airfoil surface
$\vec{V}$	Kinematic velocity
$\vec{v}$	Flow velocity
$\vec{V_a}$	Kinematic velocity of the airfoil
v_c	Tangential speed of a point on the rotating wing
V_∞	Upstream velocity
$_x$	X-direction
$_z$	Z-direction

Chapter 1

Introduction

Flapping flight is a very old concept. Inspired by the flight of birds and insects, many of the very first attempts at human powered flight, though unsuccessful, used flapping wings as the means for propulsion. The overwhelming success of fixed wing flight has rendered further attempts at manned flapping flight mostly nonexistent. Fixed wing flight has the advantages of simplicity, durability, efficiency, and safety at the large scale required to carry humans. In recent years, however, much more research is being done on small-scale flapping wing flight thanks to the push by DARPA to encourage the development of Micro Air Vehicles (MAVs) by initiating the "MAV-project" in 1997 [1]. On top of that, in the last decade urban warfare has become an increasingly important aspect of military operations, especially during the "War on Terror" in which battles are being fought in the middle of urban centers. The need for real-time surveillance inside closed buildings and other obstructed areas during such operations has increased the demand for Micro Air Vehicles that are efficient, agile, and can carry a camera and other sensing equipment. Other practical applications for MAVs include search-and-rescue missions and law enforcement [2].

However, designing such a vehicle has presented many challenges. At very small scales, traditional fixed-wing aircraft planforms suffer heavily from viscous losses. For a normal airfoil section, as Reynolds number decreases so does the max coefficient of lift and the max lift-to-drag ratio. Because of these effects, fixed-wing MAVs tend to have low performance and payload capacity. Low Reynolds number difficulties with fixed wing flight has led researchers to investigate alternative means of flight.

Flapping wings are a very intriguing alternative. Nature makes it obvious that flapping flight can be very efficient and effective aerodynamically at low Reynolds number while still allowing for a high degree of control. Hummingbirds and drag-

onflies are capable of hovering and darting deftly and quickly, while bumblebees are able to fly despite requiring a lift coefficient, in the quasi-steady sense, over twice that of any aircraft [3]. These are all highly desirable characteristics for MAVs. Much work has been done by biologists to study the flight of insects and to understand how they are able to achieve such remarkable performance since long before the DARPA initiative. Many of these studies show that insects utilize very complex kinematics patterns during a flapping cycle [4], [5], [6]. In order to develop an MAV that can rival the performance of these creatures, engineers need to utilize many of the same unsteady aerodynamic phenomena while simplifying the kinematics to a level attainable by small mechanical actuators.

1.1 Micro Air Vehicles

A Micro Air Vehicle is a special class of flying craft, defined by DARPA as having a maximum dimension of no more than 15 cm and a gross takeoff weight of no more than 100 g. There is some confusion as to the definition of an MAV because DARPA has introduced a new class—the Nano Air Vehicle—which has a maximum gross takeoff weight of 20 g, and some consider a Micro Air Vehicle to be anything smaller than a normal Unmanned Air Vehicle (UAV), which has a gross takeoff weight greater than 5 kg. For the purposes of this thesis, the original definition for an MAV will be followed. DARPA also outlines some performance guidelines which MAV designers should strive to meet. These include endurance of 60 minutes, a max speed of 15 m/s, and a range of 1 to 10 km. These guidelines are somewhat mission-dependent, however, so not all MAV types may be expected to meet all of the guidelines.

Three major MAV configurations exist: fixed-wing, rotary-wing, and flapping-wing. Fixed-wing and rotary-wing MAVs benefit from the fact that the design types are well-known and can utilize some design tools from their full-scale counterparts. However, the performance of these types is greatly affected by the flow physics in this regime, such as boundary layer transition, separation, and reattachment which may all occur within a small distance along the wing and are very difficult to predict [7]. Fixed-wing vehicles require forward speed to fly and thus cannot hover, which makes them unsuitable for many missions for which MAVs are purposed. Rotary-wing vehicles have the ability to hover and move with great agility, even at small scales, but they lose some efficiency compared to fixed-wing vehicles because their power system must provide enough thrust to support the vehicle's weight. Fixed-wing vehicles can save energy because lift is provided by the wing, and the

power system needs only to overcome the drag generated by the airframe at the given airspeed [1]. Both types of vehicle suffer at low Reynolds number due to the decreased lift-to-drag ratio (or thrust-to-drag ratio for the propeller or rotor).

Flapping-wing vehicles are similar to rotary-wing vehicles in that they can achieve a high level of control and maneuverability. There are two types of flapping-wing vehicles: bird-like *ornithopters* and insect-like *entemopters* [1]. Ornithopters actually fly more like fixed-wing aircraft in that they require forward speed to fly and produce lift by the airflow over the wings. Entemopters are more suitable for MAV tasks because they can hover and maneuver in tight spaces. Like rotary-wing vehicles, entemopters also depend on the power system to produce enough lift to hover. However, these vehicles make use of completely different aerodynamic effects in flight. Rotary-wing vehicles, though able to hover and maneuver in tight spaces, still make use of steady aerodynamics principles. As the name implies, a rotary wing can be thought of similarly to a fixed wing, but one that generates airflow over the wing not by forward speed of the aircraft but by rotary motion of the wing. Thus, rotary-wing craft are plagued by the same low-Reynolds losses that fixed-wing craft are plagued by, though the rotary motion may be used to introduce some lift-augmenting effects such as dynamic stall [6]. Entemopters, on the other hand, rely on fully unsteady aerodynamics to fly and may make use of the same unsteady and low-Reynolds phenomena that insects exploit in order to increase performance. The difficulty for designers of such vehicles is that the resulting flow field is highly complex and not well understood, despite the large amount of research done by biologists and engineers to date. Designing a Micro Air Vehicle of this type requires a large amount of time, computing power, and experimental testing.

1.1.1 The State of the Art

In the years following the original DARPA MAV-project definition, a number of of all types MAVs have been created by various firms, universities, and private designers. A small sample of these is provided here.

The Black Widow MAV is a fixed-wing vehicle. This MAV was developed by Aerovironment as a result of a study originally intended to investigate feasibility of designing an effective flying vehicle with a 6-inch wingspan [8]. This MAV was designed to maximize endurance, but not to be able to maneuver and perform surveillance in tight areas. It did, however, carry a payload consisting of a camera and control system. The final design had a 6 in wingspan (roughly 15 cm) and a mass of 80 g, and is shown in Fig. 1.1. It was able to achieve a maximum endurance of 30 min, a range of 1.8 km, and an altitude of 769 ft. However, it achieves this

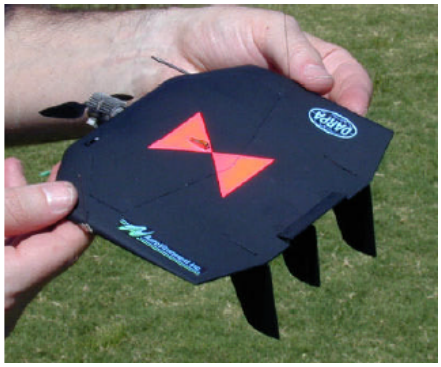


Figure 1.1: Black Widow MAV

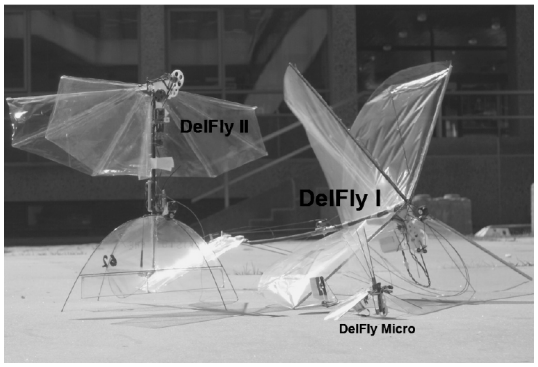


Figure 1.2: Delfly MAVs

performance while flying at about 30 mph, so it does require a considerable amount of open space to operate and certainly would not be able to hover or fly inside a building.

An example of a flapping-wing *ornithopter* Micro Air Vehicle is the Delfly. The Delfly is being developed as a project of the Delft University of Technology in the Netherlands [9]. Figure 1.2 shows the most recent iterations of the design. This MAV uses four wings in an x-wing configuration actuated by a single geared motor and crank mechanism, so the wings may not move independently of each other. Stabilization and control are provided by a system of fins and control surfaces resembling the elevator and rudder of a conventional airplane. The wings are made of very thin mylar and carbon-fiber rods, and as such are very flexible. There have been several iterations of the Delfly to date: the Delfly I, Delfly II, and Delfly Micro. The Delfly I and II are larger in size than the definition of an MAV allows, with wingspans of 50 cm and 28 cm, respectively. The Delfly Micro has a wingspan of only 10 cm and a mass of only 3.07 g. Each of these three MAVs is able to achieve forward flight by using the flapping wings to produce thrust. The Delfly II is particularly interesting because it is also able to hover by pulling the nose vertical and using the thrust as vertical force. The MAV-scale Delfly Micro is currently not able to hover, though the researchers hope to achieve this by switching to brushless motors for actuation.

A group of researchers from the Naval Postgraduate School in California designed an intriguing hybrid fixed- and flapping-wing MAV-like vehicle (it exceeds the 15 cm wingspan limit). This vehicle uses a fixed wing to provide lift and has a pair of high aspect ratio wings flapping behind the main wing to generate thrust. The flapping wings are situated inline vertically and flap towards each other to increase thrust by means of "ground effect"—the two rear wings flap in a mirror-image of each other and benefit from the same effect as a single wing flying in close proximity to the ground. The placement of the flapping wings behind the main

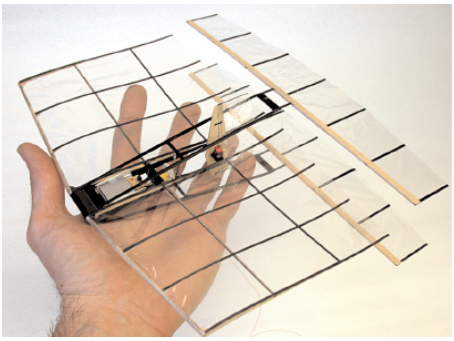


Figure 1.3: Hybrid flapping-fixed wing MAV

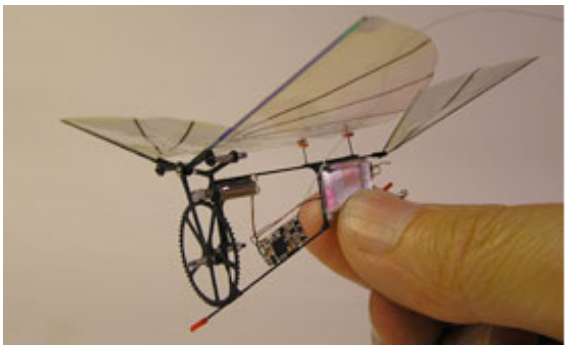


Figure 1.4: Ornithopter by Petter Muren

Figure 1.5: Nano Hummingbird

wing also has the benefit of suppressing flow separation on the main wing [10]. The smallest model produced has a wingspan of 23 cm and a mass of 11 g, though it is not fitted with a camera, and can fly for 15-20 minutes. It is shown in Fig. 1.3.

In the last decade or so, the availability of micro-sized power and actuation systems for fixed, rotary, and flapping wing vehicles has increased dramatically due to the interest in the radio control community. Small and powerful brushless motors and lithium polymer batteries have become widely available, and many MAV designs make use of these components. In fact, the radio control community itself is in some ways making great strides in the development of Micro Air Vehicles. One example is the ornithopter shown in Fig. 1.4, which won an award for being the world's smallest RC ornithopter during the MAV07 European Micro Air Vehicle Conference and Flight Competition [11]. Its wingspan is 10 cm and the mass is only 1 gram, but the flight time is less than 1 min.

Some of the very new Micro Air Vehicles show promising results have not yet been published in the literature. One example is the AeroVironment Nano Hummingbird, which can hover for 8 minutes, fly forward at 11 mph, and has precise enough control to be flown inside through a doorway [12]. It is also small enough to fit in the palm of the hand, as is shown in Fig. 1.5. The designs described above represent just a small sampling of the many that exist.

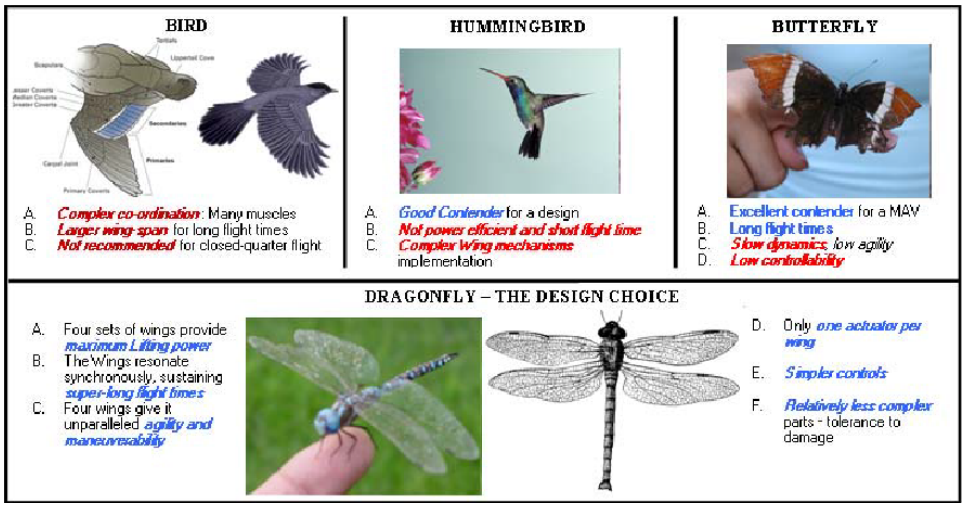


Figure 1.6: Design configuration trade study

1.2 The Quad-Wing Vehicle (QV)

This research is part of the Air Force STTR contract #FA9550-10-C-0036, and is titled "A Biologically Inspired Micro Aerial Vehicle Design and Development." A multidisciplinary group consisting of researchers and engineers from Rochester Institute of Technology, Georgia Institute of Technology, and Impact Technologies, LLC is responsible for the design of the final vehicle. As the contract title stipulates, the MAV is to be bio-inspired by emulating a flapping-wing creature. The design goals are to create a bio-inspired Micro Air Vehicle capable of hovering and precise control in order to perform surveillance and reconnaissance. Also key is that the QV must be highly energy-efficient to improve upon current MAV designs and to approach the DARPA endurance guideline. Prior to the present thesis on the design and aerodynamics of the flapping wings, a considerable amount of work has already been done on the physical configuration, control system, wing actuation, and energy management [2], [13], [14], [15].

During the early conceptual design stage, a trade study was performed to determine which natural flier the MAV should emulate. Creatures considered include the bird, the hummingbird, the butterfly, and the dragonfly. The result of the trade study is shown in Fig. 1.6 (reprinted from Ratti-et-al. [13]). Selection criteria include flight force, power, control, agility, endurance, and simplicity. The dragonfly (*Odonata-Anisoptera*) was selected because its four-wing configuration allows for good performance in many of these criteria. This design configuration is unprecendented in the MAV world.

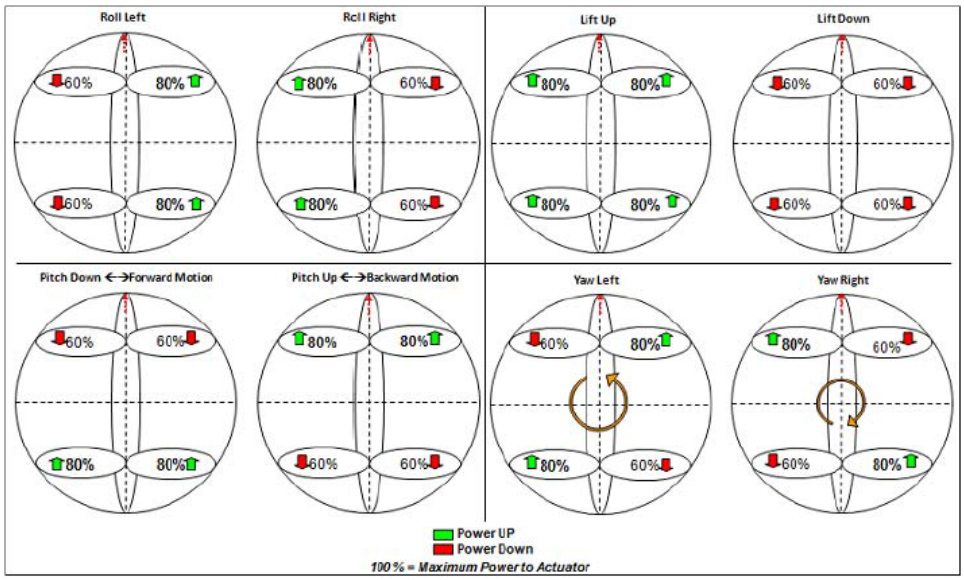


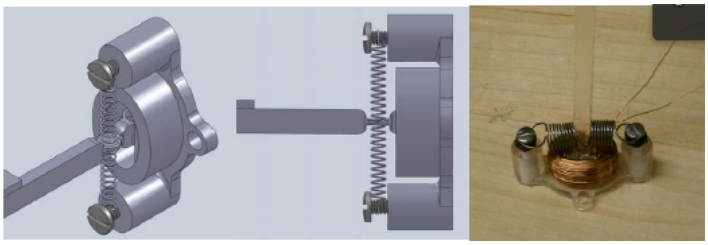
Figure 1.7: 6DOF control achieved by varying individual actuator power

1.2.1 Actuation and Control

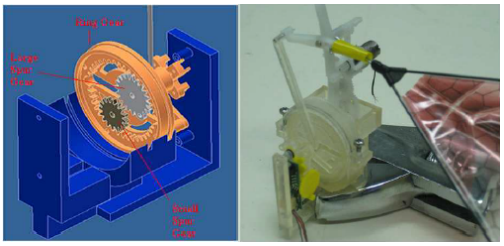
A key feature of the four-wing design of the QV is its simplicity of control. Fliers with two or four wings coupled to a single actuator require a separate means of directional control and stability. One example of this is the Delfly, which uses tail surfaces like a normal fixed wing airplane. Two-wing fliers that can move their wings independently and do not use separate tail surfaces, like the Aerovironment Nano Hummingbird, still require additional degrees of freedom in wing actuation, increasing complexity. The four-wing design of the QV, which features the ability to actuate each wing independently, is able to achieve six degrees of freedom of control (up/down, left/right, forward/backward, roll, pitch, and yaw) control by simply increasing or decreasing the power supplied to each of the four wings, as shown in Fig. 1.7 (reprinted from Ratti & Vachtsevanos [14]). By increasing power to the front wings and decreasing power to the back wings, for example, the QV will pitch up; by increasing power to one side or decreasing power to the other, the QV will roll; and by increasing or decreasing power to diagonal wings, the QV will yaw. This allows for much simpler actuation and control systems.

Flapping-wing MAVs typically vary the flapping frequency when power is applied, which is necessary when a standard motor-gearbox-crank system is used to achieve the actuation. In contrast, biological fliers typically maintain a fixed flapping frequency while varying the flapping amplitude to provide control [14]. Insects and hummingbirds achieve this by use of flight muscles that are able to flap fixed-geometry wings at the resonant frequency [16].

This actuation type presents a significant challenge for the design of a mechanical



(a) Solenoidal-magnetic actuator



(b) Hypocycloidal gear train

Figure 1.8: FiFVA actuation system

MAV, because common motors and gear systems are not designed to operate at variable amplitude. A major hurdle to the design of this MAV is the development of a miniaturized actuator that can operate in this fashion while maintaining low weight and providing high power output, though that work is not a part of this thesis. A candidate system has been proposed by Ratti et al. which features a solenoidal-magnetic actuator coupled with a hypocycloidal gear train to produce fixed-frequency, variable amplitude (FiFVA) operation [15]. The result of this work is promising, as the miniaturized system weighing 5 g was able to produce an output torque of 76 gcm at 3.6 volts and 500 mA while reaching a resonant frequency of 20 Hz. The micro solenoidal actuator and gear train are shown in Fig. 1.8 (reprinted from Ratti & Vachtsevanos [14]).

The layout of the QV sets the wings in a horizontal configuration with the wings flapping up-and-down in a vertical stroke plane. A concept sketch is shown in Fig. 1.9 (reprinted from [14]). Unlike a dragonfly, in which the two sets of wings are closely in front of and behind each other, the QV places the wings in the front and back of the body for maximum control and minimum aerodynamic interference between the two sets. With the center of mass roughly equidistant from the two sets of wings, the QV can achieve strong pitch control just by varying the power supplied to the fore and aft sets of wings, as shown in Fig. 1.7.

However, with a vertical stroke plane, simply flapping the wings in a symmetrical manner will not produce hovering force. A second rotational degree of freedom of actuation is required to introduce asymmetry between the upstroke and downstroke and produce hovering force. This degree of freedom is realized by a process called

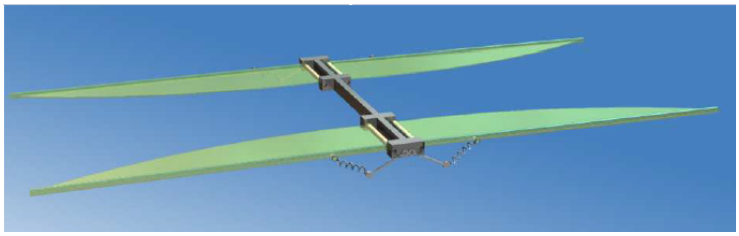


Figure 1.9: QV layout concept sketch

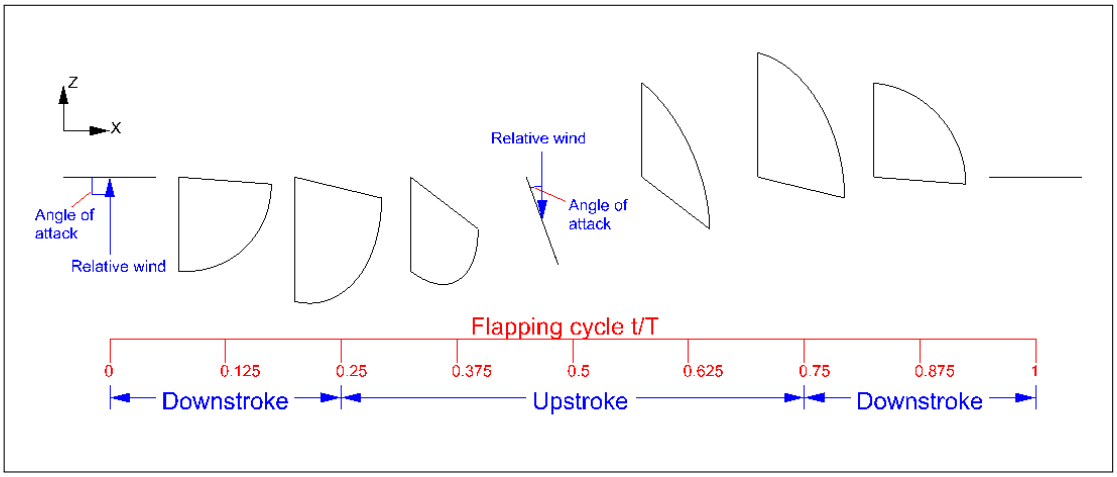


Figure 1.10: Wing motion illustration side view

feathering, whereby the wing rotates about its leading edge to vary the angle of attack between the upstroke and downstroke. The concept is that, in order to produce a large amount of vertical force for a flapping cycle, the angle of attack should be very high (near 90°) during the downstroke, but should be very low (near 0°) during the upstroke. The high angle of attack during the downstroke produces a large amount of drag on the wing, which is directed in the upward direction. During the upstroke, the feathered angle of the wing results in a low angle of attack, which ensures that there is no significant downward force produced. Most of the feathering rotation occurs at the ends of the downstroke and upstroke. Figure 1.10 illustrates an example of what a flapping cycle may look like, in the side view. In this illustration, a quarter-ellipse wing, with the leading edge on the left, is shown undergoing one full flapping cycle. The flapping amplitude for the illustration is 50° and the maximum feathering angle is 70°; this results in an angle of attack of 90° in the middle of the downstroke and 20° in the middle of the upstroke, as shown in the figure. Non-dimensional time is on the x-axis.

Feathering can be accomplished in several different ways. By introducing actuators that can supply more than one rotational degree of freedom, feathering may be achieved actively and in a precise sense. This increases complexity and weight, however. A second option, for semi-rigid wings, is to use a rotating leading edge spar

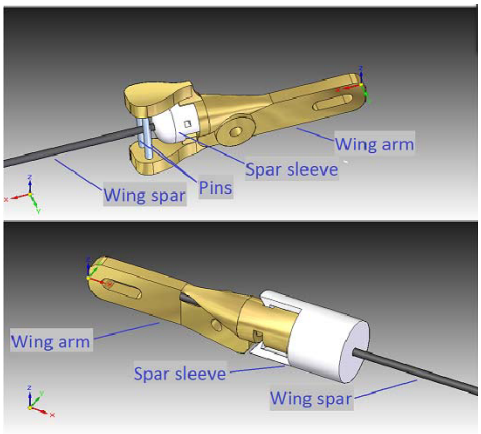


Figure 1.11: QV passive feathering mechanism

with a passive system for feathering. This option has been explored in detail by the Georgia Tech team and several designs have been tested, including the prototypes shown in Fig. 1.11 (reprinted from [14]). These feathering designs rotate about the leading-edge spar due to the pitching moments created by the aerodynamics and the wing's acceleration during the flapping cycle. These passive feathering mechanisms for semi-rigid wings may employ hard stops and/or torsional springs to achieve the desired angles for the upstroke and downstroke, as is shown in Fig. 1.11.

A different approach to the feathering problem is to introduce flexible wings instead of allowing a second rotational degree of freedom at the joint. Ho, et al. showed that heuristic optimization algorithms can be combined with a Computational Fluid Dynamics - Computational Structural Dynamics (CFD-CSD) solver to design a flapping wing that achieves high performance due to its aeroelasticity [17]. This, of course, requires access to a CFD-CSD code validated for such flow regimes and the ability to produce physical wings with material properties in close accordance with the computational model. The benefits of this approach are that the mechanical design is simplified, weight can be saved, and there are less parts to break. Because of these advantages, many insects and hummingbirds use passive aeroelastic tailoring to achieve an optimum feathering motion [16].

1.2.2 Energy Saving Design

A key benefit of flapping-wing MAVs, when designed to mimic insects or hummingbirds, is the possibility of storing flapping energy in flight to aid in actuation. Whereas a standard motor/gear/crank mechanism requires high power input to accelerate and decelerate the wings during the flapping cycle, insects and hummingbirds store energy elastically during the cycle via flight muscles [16], [18]. In fact, when a well-designed resonant system is employed on a flapping-wing MAV,

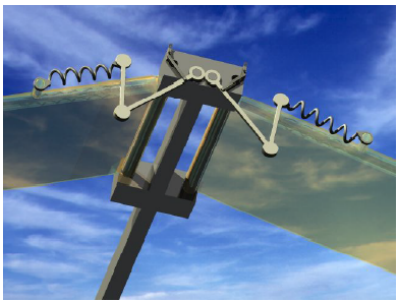


Figure 1.12: Four-bar actuation system

the inertial components of power requirement may be ignored altogether [19]; all that is required by the actuators is enough power to keep the system in a resonant state by overcoming the aerodynamic moments produced by the wings.

The QV employs a resonant flapping system similar to that found in the muscles of insects and hummingbirds, albeit simplified significantly [2]. In the simplified elastic energy storage concept for the QV, the wings are coupled to the actuators by a set of linear springs which can store elastic energy when the actuators vibrate at the system's resonant frequency. The springs may be included as part of the actuator, as is shown in Fig. 1.8(a), or they may be coupled to the wing externally as shown in Fig. 1.12 (reprinted from Ratti et al. [13]). Either way, these resonant actuation systems mimic natural fliers because the system is driven at a fixed frequency, with control power provided by varying the amplitude.

In addition to the energy savings provided by driving the flapping wing system through a resonant elastic system, Ratti and Vachtsevanos have demonstrated that increasing the number of wings (and actuators) can save energy [2]. The percentage of energy savings for the entire vehicle is dependent on the ratio of power required by the actuators to the total power required by the vehicle, which also includes an avionics system and controller.

1.2.3 Required aerodynamics research

One major point in the design stage for the QV is the design of the wings. Given the definition of an MAV, the maximum wing dimensions are already set. More important, for the design of the QV, is determining how to actuate the wings to achieve the best performance. Specifically, since the major goals of the QV are to be able to hover and move precisely in tight locations, such as inside buildings, the initial stepping stone is to achieve hovering flight. Fast forward flight is not as important relative to the goals of this craft and the maximum forward speed of the aircraft will be a function of the flapping frequency and wing dimensions [18].

In accordance with the aforementioned mission goals for the QV, the major aerodynamic research required for the QV is the estimation of hovering force available, power required, and actuation system requirements for hovering flight of the QV. Specifically, given a target flying weight, the team needs to know the resonant frequency required by the actuation system, the corresponding flapping angle to achieve hovering flight at this frequency, and the feathering characteristics that provide the best performance at this design point. Other important takeaways from the research should include the effect of scaling the wing, the possibility of interference effects between the fore and aft wing sets, and the optimal stroke plane orientation for hovering flight.

Most of these research requirements can be performed through the use of computer software. Efforts within the team are also in place to develop reliable experimental benchtop systems to test wing and actuation designs, to simulate the dynamics of the QV, and to validate the computational results. From the experimental aerodynamics side, before any testing and validation can be performed, it is first necessary to develop methodology and hardware that allow the capture of aerodynamic forces generated by flapping wings, primarily in hover-mode. This is no small task, given the relatively small aerodynamic forces generated by MAV-size flapping wings and the relatively large inertial and vibratory responses to the high flapping frequency required for hovering flight. Once a reliable system is developed, experimental tests can be performed to validate the computational results and potentially to test new designs in less time than is required for the computer simulations.

1.3 Design Tools

Aerodynamic design of a Micro Air Vehicle, and especially a flapping-wing Micro Air Vehicle, is a very challenging task. Because the flow is highly unsteady and characterized by flow separation and vortex shedding, many tools and assumptions useful for traditional steady high Reynolds number aerodynamic study are no longer applicable. Most studies published in the literature use either a full Navier-Stokes numerical solver to analyze the flow or an experimental testing setup. While these tools can be very useful, they significantly increase the time and effort required to get even an order-of-magnitude prediction of the forces and moments acting on an MAV wing. Even using these tools, obtaining valid results is challenging because the flow is very complex. Any computational method used to design flapping wings of an MAV must be validated in a low-Re, high-vorticity regime in which these wings

operate. Experimental testing is also difficult, in part due to difficulties already mentioned related to the relative magnitude of the aerodynamic forces produced by a wing versus the inertial forces, and partly because accurately duplicating the complex flow patterns produced by a hovering MAV in a simulation environment is not always possible. This section describes the available design tools for flapping MAVs.

1.3.1 Computational Fluid Dynamics

Computational Fluid Dynamics, or CFD, is a very powerful analysis tool that has become extremely popular in recent years due to advances in computing speed. CFD is a numerical method to solve the Navier-Stokes equations, which are a system of coupled partial differential equations applying Newton's Second Law of motion to a fluid. When required, CFD solvers also solve the energy equation simultaneously with the Navier-Stokes equations. Depending on the flow regime, simplifications can be made to reduce computational complexity. Such simplifications include incompressible flow (constant density), steady flow (time independent), laminar flow (no turbulence), and inviscid flow (no shear stresses). The basic output produced by a CFD code is a velocity and pressure field, by which other quantities such as forces, moments, flow rates, and power requirements can be computed.

The numerical process employed in CFD solvers is discretization of the flow equations, often by use of a Taylor series approximation. The region on which the equations are solved is called the *fluid domain*. In order to solve the flow field on the domain in a discrete sense, the domain itself must be discretized in addition to the flow equations. The process of discretizing the fluid domain is called *meshing*, as the final result is a mesh, at the nodes of which the flow equations are solved. The quality of the mesh, and especially the spatial distance between mesh nodes, may greatly affect the computational solution because the error in the Taylor series expansion depends on this distance. Additionally, the resolution of the mesh usually is not uniform because the flow physics require more resolution near boundaries, in wakes, and around complex geometric features. Meshing itself can be a very difficult and time-consuming because of these concerns, and especially if moving boundaries are involved as is the case with flapping wings.

CFD is a very powerful design tool because it can solve the flow field while making very few limiting assumptions. However, it also requires a large amount of time, both for setting up the mesh and in computation. Each time the equations are solved, they must be solved for thousands or even millions of points in the domain, depending on how large the mesh is. When the Navier-Stokes equations

are solved, due to the complexity of the equations they are usually solved in an iterative manner, further increasing the computational time. Additionally, in a time-marching procedure for an unsteady flow, the entire domain must be solved over again for each time step.

1.3.2 Other computational and analytical methods

Because solution of the Navier-Stokes equations is such a time-consuming process, engineers often look for other computational tools in order to speed the design process. These tools usually simplify the problem by making some important assumptions. Under the appropriate conditions, these assumptions are valid and the simplified equations or methods will be suitable to provide insight into the problem at hand.

For flapping Micro Air Vehicles, the complexity of the flow field makes it difficult to find simplified models that can still produce useful insight. Researchers have used different approaches to model flapping flight of MAVs and small biological creatures such as insects and hummingbirds. Some examples of these methods are quasi-steady approaches combined with blade-element theory [4], [5], [20], analytical models based on linear theory [21], [22], analytical models based on nonlinear theory [23], minimum loss theory (adapted from propeller analysis) [24], and numerical panel codes based on potential flow theory [10], [24], [25], [26]. Each of these methods has its advantages and disadvantages and must be used with care and knowledge of the governing assumptions that it employs.

1.3.3 Experimental methods

A third way to analyze the aerodynamics of flapping Micro Air Vehicles is through experimental testing. Experimental testing for flapping wings involves, at the least, a model wing, actuation system, and measuring device. Measuring devices are usually load cells or strain gauges. Systems for testing an MAV in forward flight must also make use of a low-speed wind tunnel. More advanced setups may also include means of flow visualization such as particle image velocimetry.

Experimental analysis of flapping wings presents a unique challenge because the forces measured by the recording device are not only due to aerodynamic effects. Especially for MAV sizes where the flapping frequency and amplitude are very high, inertial loading makes up a significant portion of the force traces. There are various ways of isolating the aerodynamic forces in an experimental setup. One is to estimate the inertial loading based on the kinematics and inertial characteristics

of the wing [27]. This can be very difficult, especially for wings that are flexible and of unusual shape. Additionally, the vibration of the mounting system on top of the measuring device contributes to the force trace and would be nigh on impossible to predict accurately. A second method of isolating the aerodynamic force is to do it experimentally by subtracting off the measured vacuum force trace [28]. By testing the same experimental model in a vacuum, all effects besides the aerodynamic ones are removed and the aerodynamic forces can be measured with greater certainty. A third method is to use the concepts of dynamic similarity to create an experimental setup that is much larger and moves more slowly. This can be achieved by immersing the wing in a fluid other than air; Sane and Dickinson used oil [4], [5]. Using this concept, Sane and Dickinson were able to lower the actual frequency in their experiments to 0.17 Hz, though the flow field was still representative of that around a fruit fly's wing, rendering the inertial forces inconsequential.

If the experimental testing is performed in air, special care must be taken in selecting a measuring device. It must be able to accurately measure the relatively low-magnitude forces produced by an MAV-size flapping wing while also being able to handle the higher-magnitude inertial and vibratory loads. Because of this requirement, it is necessary to minimize the weight of the hardware on top of the measuring device so as not to overload it, and careful attention must be paid to the design of the experimental system.

Chapter 2

Literature Review

The concept of using flapping wings for flight of a Micro Air Vehicle is a relatively new one in aviation history terms. Successful hovering flapping flight of MAVs is only now beginning to be realized because of advances in miniaturization of electronics and power systems. As such, the body of work in the literature relating to flapping MAVs is still quite small.

However, MAV designers can borrow from unsteady aerodynamics research from other fields, as many of the same concepts apply to flapping MAVs. Perhaps the greatest source of information for designers of flapping MAVs is the research done by biologists in the last 30 or 40 years in the study of birds and insects. Especially in recent years, many studies have been done to observe the wing kinematics, through use of high-speed cameras, of hovering creatures such as insects and hummingbirds and to duplicate them in a mechanized setup or computer simulation. Some concepts and analysis tools can also be borrowed from unsteady aerodynamics applied to flutter and other phenomena that have been of interest to engineers for decades.

2.1 Analytical Work

The first analytical work on unsteady aerodynamics found in the literature was performed by Theodore Theodorsen in 1935 [22]. Theodorsen's work applied to two-dimensional wing sections or wing-aileron combinations undergoing flutter, with the goal of predicting aerodynamic instability in these conditions. This work was of particular interest to the United States military during World War II as aircraft became faster and the flutter phenomenon became very important. Theodorsen's theory was based on potential flow and considered a wing or airfoil undergoing oscillations of very small amplitude. Due to these constraints, the theory cannot be directly applied to the hovering flight of Micro Air Vehicles, but many of the

same concepts were used by later researchers to develop theories. Theodorsen's functions as well as the findings of later researchers such as Garrick, however, can be applied as a good descriptor of the flight of larger vehicles and creatures that have a relatively fast forward flight speed and wing flapping amplitude [29].

Theodorsen's work was mostly concerned with the lift force. Subsequently, Garrick extended his work to predict the horizontal thrust force generated by an airfoil or aileron flapping or oscillating about an axis parallel to the wing span [30]. Garrick cited von Kármán and Burgers [31], whose work identified a street of vortices in the wake which are responsible for the horizontal forces produced. Garrick's paper was also based on potential flow theory and considered the wake to be infinitely thin, but allowed for flapping oscillations of finite amplitude. Garrick presented two different methods to calculate the horizontal force, the first based on the energy in the wake resulting from the vortex street and the second based on the pressure distribution on the airfoil. More recent researchers using numerical panel codes based on potential flow theory often compare their results with Garrick's theory for validation and to compare more realistic wake representations with the infinitely-thin wake assumption used by Garrick [32]. The works of Theodorsen, Garrick, von Kármán, and other early researchers to study unsteady aerodynamics are explained in greater detail by Azuma, et al. [19].

A number of extensions of the original work of Theodorsen, Garrick, von Kármán and Burgers, and others have been performed in the years following, with the goal of developing an analytical model of insect flight. Zbikowski proposed an analytical model of insect flight based on two approaches: one based on the circulation methods of von Kármán and Sears [33] and another based on the velocity potential [34]. Zbikowski's approach put the focus on two major phenomena experienced in insect flight: the nonlinear leading-edge vortex and the linear attached flow portion. His theory was linear with nonlinear corrections applied for the leading edge vortex, and was based on potential flow. Pedersen used a method similar to Zbikowski's velocity-potential treatment to predict the flight forces on insect wings in hover and achieved good results for the lift force [35]. Minotti used a similar approach, but based his methods on the circulation approach. The leading edge vortex was modeled by a single discrete vortex shed from the leading edge [36]. His work applied to two-dimensional wing sections, but was extended to three dimensions by a simple factor of the wing tip velocity and the square root of the non-dimensional second moment of wing area. Minotti compared his theoretical prediction to the experimental results of Sane and Dickinson [4].

Ansari developed a novel analytical approach to the problem of insect-like flap-

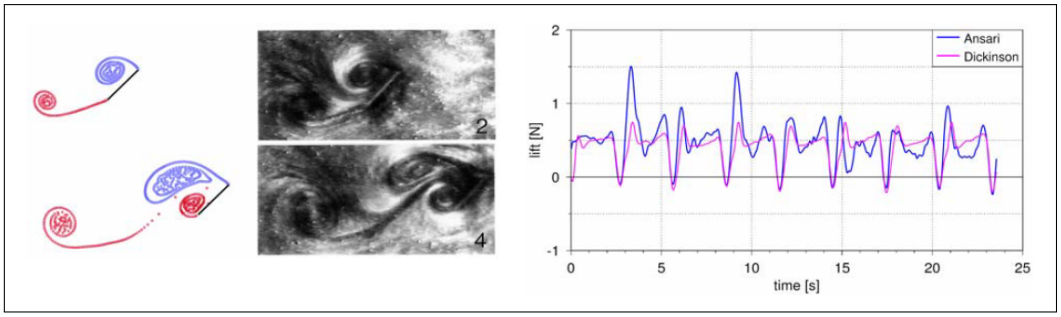


Figure 2.1: Ansari's method compared with experimental results

ping flight [23]. His approach was based on a set of coupled nonlinear wake-integral equations resulting from the treatment of the wing section as a flat plate with leading-edge and trailing-edge separation. His theoretical approach assumed potential flow, though it was noted that the separation at the leading and trailing edges, as well as the Kutta condition employed, are due to viscosity in a real flow. His approach was "quasi-three-dimensional" in that the two-dimensional theory was extended to three dimensions via blade element theory with radial chord lines. The governing equations developed were solved using numerical methods because there is not a closed-form solution. Remarkable agreement was found between Ansari's analytical method and experimental results both for the flowfield visualization supplied by Dickinson and Götz [37] and the prediction of lift supplied by Dickinson, et al. [38], as can be seen in Fig. 2.1 (reprinted from Knowles, et al. [39]). Ansari did note the limitations of his method, perhaps most importantly its inability to capture three-dimensional flow effects such as tip vortices.

2.2 Computational Work

Various computational studies on flapping-wing flight of insects, hummingbirds, and MAVs have been performed. The majority of studies involving hovering flight are performed using CFD due to its ability to predict and effectively capture the flow physics. Other computational studies use a variety of methods to overcome the high computational cost and time requirement for CFD.

Biologists have done much research into the hovering flight of insects and hummingbirds. This type of research is analysis-oriented, not design-oriented, in that its goal is to determine exactly how these creatures are able to fly with the hope that the insights will be valid for future researchers or engineers. Thus, most biologists devote much effort to recording and detailing the complex wing geometry and kinematics of these creatures.

One such study was performed by Liu and Sun, who observed hovering droneflies

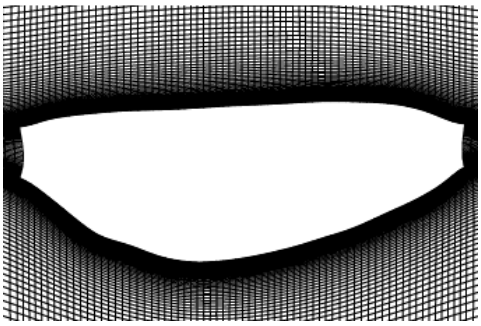


Figure 2.2: Dronefly wing mesh used in [40]

[40]. In this study, the wing kinematics of a dronefly were observed in three dimensions using three high speed, orthogonally-mounted and synchronized cameras. Liu and Sun also measured the shape of the wings and morphological parameters of the insect body, including its mass and center of mass. They then simulated the hovering kinematics of a single dronefly wing in a CFD environment (wing-wing interactions and wing-body interactions were ignored). The CFD code solves the non-dimensionalized incompressible Navier-Stokes equations with the assumptions that the wing is rigid and the flow is laminar. The wing mesh is a conformal O-H grid and is shown in Fig. 2.2. The CFD code returned an average vertical force for a single cycle approximately equal to the weight of the dronefly, and the horizontal force and the moments were balanced, providing validation for the CFD model.

Sun and Tang performed a similar study, but the modeled insect was a fruit fly [41]. Sun and Tang used the same CFD code as Liu and Sun [40] and also used a very similar wing mesh. In this study, instead of trying to match the natural wing kinematics exactly, the wing motion was idealized and some of the kinematics parameters were varied. Key findings of this study include the result that the rotation-advanced case—where wing rotation occurs mostly before stroke reversal—results in greater vertical force. Sun and Tang hypothesized that changes in wing rotation timing and the magnitude of angle of attack may be used by flying insects to produce greater force for maneuvering.

The concept of a leading-edge vortex has already been discussed as it relates to recent attempts to model insect flight analytically. The phenomenon was first observed experimentally by Ellington, et al. while simulating the kinematics of a hawkmoth on a benchtop flapping device with smoke for visualization [42]. Since then, much interest has been paid to the leading-edge vortex because it is believed to be one of the major contributors to the lift generated by the wings of insects in hovering flight. The leading-edge vortex observed on the wings of insects is actually very similar to that observed on delta-winged aircraft, which is caused by separation

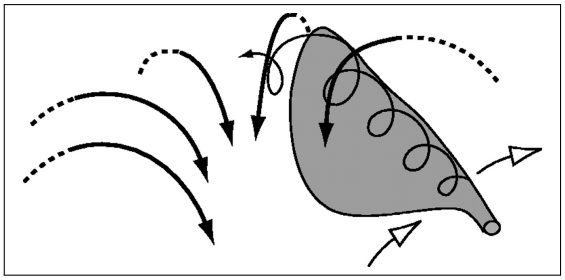


Figure 2.3: Leading-edge vortex from [21]

and is strong and stable and augments lift by creating suction on the top surface of the wing [43]. The suction zone allows the flow to reattach downstream of the vortex, even when the wing is experiencing a high angle of attack [21]. Sane also noted that the interaction between the leading-edge vortex and the spanwise flow near the wing tip creates a strong downwash behind the wing, as is shown in Fig. 2.3 (reprinted from Sane [21]). Several researchers have found that the leading-edge vortex also arises from the Navier-Stokes solution of the flow field in flapping-wing simulations, so CFD can be used to investigate this phenomenon.

In a paper on the aerodynamics of hawkmoth flight, Liu and Kawachi focused on the leading-edge vortex and its role in unsteady lift enhancement during flapping flight [6]. Using CFD and a computational model of a hawkmoth wing, they simulated flapping flight and observed the structure of the leading-edge vortex. Liu and Kawachi also observed that the leading-edge vortices develop a very strong axial flow in the core as they increase in size, and noted that this axial flow is the major contributor to the stability of the vortex; the vortex quickly breaks down near the wing tip where axial flow on the top of the wing is hindered by an adverse pressure gradient. They also suggested a link between the leading-edge vortex and the dynamic stall effect observed in rotary-wing vehicles. Liu and Kawachi reported that approximately 80% of the vertical force for the flapping cycle is produced when the leading-edge vortex is strongly attached to the wing, lending credence to the notion that it is largely responsible for the high vertical forces produced in hovering flight.

In computational studies of hovering insect flight, typically only the wings (or just a single wing) are modeled and the body is ignored. Additionally, researchers investigating the vertical hovering force produced by the wings are generally only concerned with the average lift produced during a flapping cycle, and the oscillations of the body due to periodic changes in forces during the flapping cycle are ignored. In order to investigate whether these oscillations are actually important, Wu, Zhang, and Sun simulated the hovering flight of insects using CFD coupled with the equations of motion [44]. They considered two different insect species: the

dronefly and the hawkmoth, which are on opposite ends of the insect spectrum in terms of size, flapping frequency, and wing-mass to body-mass ratio. They found that mainly horizontal oscillations occur during the flapping cycle, and they serve to slightly increase the power required to sustain hovering flight. This effect is larger for the hawkmoth than for the dronefly (9% increase vs. 2%) and can be ignored for the study of most insects.

Ho, et al. presented a study on flapping wings at MAV-size focusing on the effect of wing deformation [17]. Though most studies found in the literature make the assumption that the flapping wing is rigid, in reality the wings of insects and hummingbirds deform, in some cases severely, in flight. However, the study of aeroelasticity in flapping flight requires precise knowledge of the material properties of the wing and a way to either simulate the deformation computationally or to reproduce it accurately in a physical model. Ho, et al. did both, and applied an optimization algorithm to maximize the net lift and thrust produced during a flapping cycle by varying the elastic properties of the wing. The study showed that, by carefully designing the wing's elastic properties, significant gains in performance can be made over similar rigid wings. The work of Liu and Kawachi [6] also implemented deformable wings.

Ashraf, et al. presented a review of computational and analytical studies of flapping-wing aerodynamics [45]. Though most of the work presented in the paper focuses on using flapping wings for propulsion, not lift in insect-like hovering flight, it still contains practical concerns for this flight regime. In the paper, the researchers also performed 2D simulations of their own using the FLUENT CFD solver, varying the values of two non-dimensional parameters related to flapping flight over a broad range. They found that the propulsive forces generated follow the trend predicted by Garrick's linear theory, even though in the cases of very high unsteadiness (closer to the realm of insect flight) leading-edge vortices were observed. They also found that in the low-Reynolds number range studied, the laminar flow assumption is valid. This assumption is supported for insect flight throughout the literature [6], [23], [34].

While CFD is the most widespread computational method used in studying flapping-wing flight, a number of researchers have used other computational tools as well. Jones and Platzer used a vortex panel code to evaluate thrust production of flapping wings in various configurations [32]. They also compared the results of their code to the theoretical predictions of Garrick and found good agreement, although for larger degrees of unsteadiness the planar wake assumption made in the linear theory fails to capture some effects. Using the same code, Jones, et

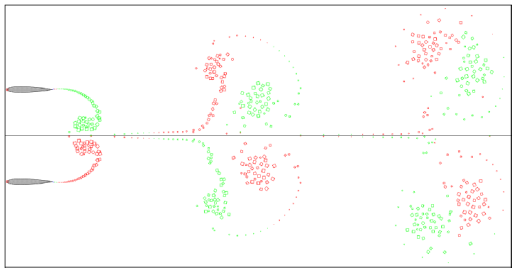


Figure 2.4: Flapping in ground effect from [46]

al. simulated airfoils in ground-effect using the method of images [46] and applied their work to a Micro Air Vehicle, which can be seen in Fig. 1.3 [10]. The wake structure is shown in Fig. 2.4 (reprinted from [46]). Jones, et al. also compared their results to experimental and CFD simulations for flows with relatively low degrees of unsteadiness and achieved good results [25]. However, their simulations were not for hovering motion and their code was not validated for that regime of flight.

Hall and Hall used minimum loss flapping theory to determine the optimum wake circulation distribution to produce thrust in forward flight for flapping MAVs and birds [24]. The study involved large-amplitude flapping oscillations at both high and low Reynolds numbers. As part of this work, Hall and Hall also investigated the effect of wake roll-up using a 2D vortex panel code. Wake roll-up is the induced motion of the wake resulting from time-varying forces in the unsteady flow. Alternatively, the wake may be *prescribed*, meaning that vorticity in the wake is assumed to remain in the same place it was originally shed, though it is convected downstream with the surrounding fluid. For an oscillating wing or airfoil, in a real flow the wake rolls up and expands away from the shedding plane in a von Kármán street [47], as shown in Fig. 2.5. In contrast, a prescribed wake behind an oscillating airfoil or wing resembles a sinusoid. Hall and Hall found, for their purposes, that a prescribed wake representation was sufficiently accurate.

Smith, et al. also used a vortex panel code to calculate the forces generated by flapping wings [26]. This study is unique in that it uses a three-dimensional panel code and models the aerodynamics of an insect (the sphingid moth) in flight. The results were compared with experimental measurements and quasi-steady predictions and showed improved accuracy over the quasi-steady methods.

A final computational method which is fairly popular in the literature is the quasi-steady analysis extended to three dimensions using blade-element theory [4], [5], [20], [21], [26]. However, this method fails to account for unsteady effects that occur in hovering flight, including the leading-edge vortex. Ellington showed,

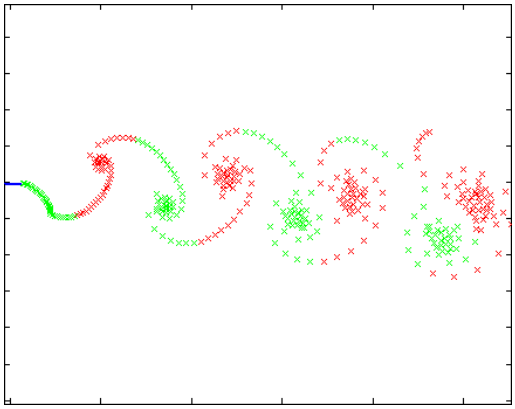


Figure 2.5: von Kármán street

in 1984, that the existing quasi-steady models of insect flight were insufficient to explain how insects achieve enough lift to hover [48]. Nevertheless, due to the simplicity and low computational cost of the quasi-steady analysis, later researchers continued to use it and several updates to the model were implemented to improve its accuracy, such as the inclusion of rotational effects by Sane and Dickinson [5].

2.3 Experimental Work

Experimental studies on flapping wing aerodynamics of birds an insects are fairly common in the literature, especially by biologists. As early as 1973, Weis-Fogh used high-speed cameras to capture the wing kinematics of small hovering insects such as the wasp *Encarsia formosa* and proposed the "clap-and-fling" mechanism by which some insects are thought to possibly increase hovering force by making use of interactions between the wings on either side of the body [49]. However, Sane notes that many insects do not ever clap their wings together, so the effect cannot be thought of as a major lift-enhancing device for most insects; Sane also notes that the "clap" may simply be a result of the insects maximizing their flapping amplitude [21]. Weis-Fogh's high-speed cameras were also not fast enough to capture the entire wing cycle kinematics with complete accuracy.

More recently, measuring devices and approaches, high-speed cameras, and flow visualization techniques have all improved, lending themselves to a great increase in the number of experimental studies on flapping-wing flight. Sane and Dickinson took a novel approach to measuring the flight forces on the wings of the fruit fly by using a dynamically scaled model immersed in oil [4], [5]. Dynamic scaling was achieved by matching the Reynold's number, which for this case was on the order of 10^2. Sane and Dickinson varied several kinematic parameters systematically and observed the effect on the forces and aerodynamic efficiency, while also comparing

their results with quasi-steady predictions. Their dynamic-scaling setup is particularly beneficial because, by scaling up the wing and increasing the viscosity of the working fluid, the magnitude of fluid forces is increased many times, making them much easier to measure. Dynamic scaling also allowed them to decrease the flapping frequency to 0.17 Hz, which greatly reduces the inertial loading due to rotational acceleration.

Lentink and Dickinson implemented a similar experimental setup using the principle of dynamic scaling to gain insight into the mechanisms responsible for stable leading-edge vortex attachment in insect flight [50]. Their materials and experimental methods were similar to those used previously by Sane and Dickinson, though they also used air bubbles released from the leading edge of the wing to visualize the leading-edge vortex, as shown in Fig. 2.6 (reprinted from [50]), and they used the same scaled fruit fly wing. A key finding of their experiments was that the Rossby number, which is the ratio of inertial forces to the Coriolis force, is largely responsible for the stability of the leading-edge vortex. They found that wings, or regions of wings, with a Rossby number on the order of 1 will allow for stable leading-edge vortex attachment. Since the Rossby number for a full wing is directly proportional to aspect ratio, it follows that wings with a low aspect ratio should be used to best utilize leading-edge vortices.

Lentink and Dickinson also found that, while the non-dimensional amplitude of flapping did not affect the *stability* of the leading-edge vortex, it did affect the *integrity*; larger centripetal accelerations due to high flapping amplitude can prevent spiral burst of the leading-edge vortex. Spiral burst is a phenomenon observed in delta-wing aircraft at high angles of attack and is related to large-scale instabilities in the vortex [51]. Lentink and Dickinson found that spiral burst does not affect the force-augmenting effect of the leading-edge vortex, and, quite interestingly, the Reynolds number does not seem to play a role in the behavior of the leading-edge vortex either. They report that this finding is consistent with phenomena observed in large wind turbines, where greater-than-expected forces have been observed on the inner part of the blades which experience high Reynolds number, high angles of attack, and low local Rossby number.

Since the leading-edge vortex was first discovered on insect wings, it has often been speculated that it is exclusively a low-Reynolds-number phenomenon. Recently, however, several studies have shown that it is present even at larger scales. The study of Lentink and Dickinson presented above is one. Another example is the work of Hubel and Tropea , who constructed a flapping setup allowing for the measurement of forces and used particle image velocimetry to observe the flow pat-

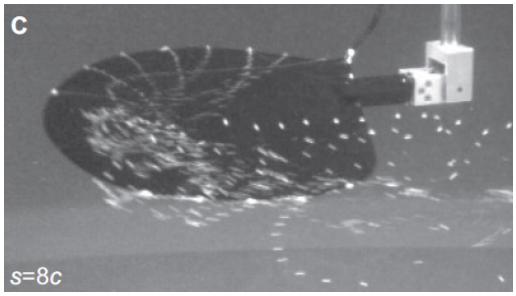


Figure 2.6: Leading-edge vortex visualization from [50]

terns on bird-size wings in "normal" flying conditions [52]. They found that even in these conditions, for which quasi-steady methods are normally used to approximate the flight forces, leading-edge vortices can be observed.

Mayberry and Lehmann constructed dynamically-scaled models of dragonfly wings and measured the forces acting on them [53]. The study modeled both the forewing and the hindwing and investigated the effect of varying the flapping phase relationship between the two wings. They found that, depending on the values of certain kinematic parameters, the aerodynamic interactions between the two wings could be quite important, and in most cases decreased the overall lift generated by the two wings, and especially that of the hindwing. However, the hindwing was found to gain performance near to that of the forewing when its motion led that of the forewing by one-quarter stroke (90° phase lead). The interaction between the two wings was found to depend greatly on the separation between them, and at 5 chord-lengths away aerodynamic interactions between the two wings were relatively unnoticeable.

Other notable experimental studies of flapping wings include those of Jones, et al. [54], who simulated their flapping-wing Micro Air Vehicle in an experimental setup and by using various numerical methods; Ho, et al. [17] who used flexible MEMS wings with active deformation control to affect the flow patterns around the wing and to improve the aerodynamic performance; Frampton, et al. who used passive aeroelastic tailoring to optimize thrust production in one-degree-of-freedom flapping motions [55]; Ames, et al. [27] who measured the forces and the flow field (using particle-image velocimetry) on a flapping rectangular wing and compared the results to those of an unsteady panel code; and Bomphrey, et al. [3] who visualized the leading-edge vortices on bumblebee wings and determined that the left and right wings developed independent vortex structures.

2.4 Statement of Work & Literature Gaps

The major focus of this thesis is to perform the aerodynamics research required for the design of the QV, as detailed in Section 1.2.3. The research is performed in three sections: Computational Fluid Dynamics analysis to study the aerodynamics of the flapping wings and to determine the kinematics parameters for best aerodynamic performance; the application of an unsteady 2D panel code based on potential flows to determine whether this method is viable for Micro Air Vehicle design; and design of an experimental benchtop setup to measure forces on the flapping wings and to attempt to isolate the aerodynamic forces.

1. The Computational Fluid Dynamics analysis includes the following activities:

    - Grid independence study
    - Validation of the CFD code against data in the literature
    - Kinematics parameter sweeps to maximize lift and minimize power consumption
    - Investigation of flow field characteristics
    - Investigation of the effects of wing scale
    - Two-wing simulation
    - Flapping with a horizontal stroke plane, to compare with the vertical stroke plane of the QV

2. The potential flows analysis includes the following activities:

    - Development of a 2D unsteady panel code for airfoils
    - Validation of the panel code against data in the literature
    - Definition of non-dimensional parameters for 2D hovering flight
    - Comparison of the code against CFD for hovering flight of an MAV
    - Discussion of comparison results
    - Recommendations for using panel codes as a design tool for hovering, flapping MAV flight

3. The experimental work includes the following activities:

    - Design of a benchtop setup to measure the forces on flapping wings
    - Construction of a set of flapping wings for testing purposes

- Interfacing the benchtop setup with a digital computer to record data
- Investigation of methods to isolate aerodynamic loads, including vacuum-chamber testing
- Estimation of inertial loads
- Recommendations for the design of benchtop setups to test flapping wings for MAVs experimentally

In addition to being required for the design of the QV, the statement of work proposed above fits well with gaps in the literature review. On the CFD side, there is no known CFD study found in the body of literature that systematically sweeps kinematic parameters to optimize the performance of a flapping-wing MAV in hovering flight. In fact, the only work encountered in which CFD was used specifically to study the aerodynamics of a flapping-wing Micro Air Vehicle was performed by Jones, et al. [25], but their MAV did not hover or perform insect-like flapping. This thesis will contribute significantly to the body of literature by detailing the aerodynamic effects of flapping kinematics parameters, by determining whether unsteady force-augmentation phenomena found in insect flight, such as leading-edge vortices, can be expected in the flapping flight of Micro Air Vehicles as well, and by determining whether the somewhat uncommon horizontal stroke plane used in the QV design is viable for hovering flight.

Though unsteady panel codes are somewhat common in the study of flapping wings, they have very rarely been applied to hovering flight. This thesis investigates whether these codes can be used for the design of a flapping Micro Air Vehicle in hovering mode by comparing it with CFD over a range of non-dimensional parameters. The results of this study allow MAV designers to know whether unsteady panel codes can be used in the design of such vehicles, and if so, under what conditions. Recommendations are also made to improve the performance of unsteady panel codes. The high computational cost and setup time of CFD provides the motivation for using simplified computational methods, such as unsteady panel codes, for the design of Micro Air Vehicles.

On the experimental side, a fair number of studies of hovering flapping flight have been done over the years, especially by biologists. Many of these have been quite successful and have provided valuable insight into the forces produced by hovering creatures and into the aerodynamic phenomena present. Many of these tests have used scaled-up models that required very low flapping frequencies. In this thesis, the experimental setup was designed to test the actual-scale wings of a flapping MAV in air, and the analysis focuses on the isolation of aerodynamic

forces from other forces, such as inertial loading and vibration, that are also a part of the measured force. Though some experimental tests of actual-scale MAV wings have been performed (see [20], [27], [28], [55]), very little attention has been paid to isolating the aerodynamic forces from the other loads. This thesis makes recommendations on how to isolate the aerodynamic forces.

Chapter 3

Computational Fluid Dynamics Analysis

Computational Fluid Dynamics (CFD) is a numerical method to solve the partial differential equations governing fluid flow, including conservation of mass (continuity equation), conservation of momentum (Navier-Stokes equations), and conservation of energy. These equations need to be solved simultaneously, and, due to their complexity, are impossible to solve analytically for all but the simplest of cases. CFD is a very powerful tool because it is able to solve the equations approximately on a discretized domain (known as a mesh or a grid) using numerical methods. CFD is extremely useful for engineers because it can be applied to almost any fluid flow problem, and it can also be coupled with other physical models such as the equations of rigid-body motion, solid deformation, heat transfer, combustion and other chemical reactions, evaporation, etc. This usefulness does come at a price, however. CFD is very computationally expensive because of the complexity of the governing equations and the usually large number of discrete points in the mesh on which it is solved, and creating the mesh also usually represents a significant time investment for the user.

In this chapter, ANSYS FLUENT 12.1, a commercially-available CFD code, is applied to the problem of hovering flapping flight of a Micro Air Vehicle. The results give insight into the forces, moments, and power requirement that can be expected in hovering flight of the Quad-Wing Vehicle, as well as identify performance requirements for the actuation system.

3.1 FLUENT Solver

As mentioned above, ANSYS FLUENT 12.1 is a commercially-available CFD solver. It is chosen for this work for several reasons:

1. It has the ability to simulate arbitrarily translating and rotating bodies, as is the case for flapping wings.
2. It has been used in the literature to study flapping flight [45].
3. It has been validated against data in the literature for this regime of flight (see section 3.2).
4. The solver (FLUENT) and mesh generation program (GAMBIT) are available at Rochester Institute of Technology.

FLUENT has the ability to solve the governing equations of fluid flow in a very general form; based on the problem at hand, simplifications can be made. The general form of the continuity equation solved by FLUENT in an inertial frame is given in Eq. 3.1. Equations 3.1, 3.3, and 3.4 are published in Chapter 18 of the FLUENT 12 Theory Guide [56].

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = S_m \tag{3.1}$$

In Eq. 3.1, S_m is a source term applied for problems in which mass is entering the continuous phase from another phase or to model user-defined sources. For the present analysis, there are no such sources and the Mach number is low; thus, the source term is dropped out and the flow is assumed to be incompressible (density is constant), as given by given by Eq. 3.2.

$$\nabla \cdot \vec{v} = 0 \tag{3.2}$$

The general form of the momentum conservation (Navier-Stokes) equation is given by Eq. 3.3.

$$\frac{\partial}{\partial t}(\rho \vec{v}) + \nabla \cdot (\rho \vec{v} \vec{v}) = -\nabla p + \nabla \cdot (\overline{\tau}) + \rho \vec{g} + \vec{B}_{other} \tag{3.3}$$

In the above equation, $\rho \vec{g}$ is the body force due to gravity and $\vec{F}$ is a vector of other body forces that may arise. $\overline{\tau}$ is the stress tensor, given in the Eq. 3.4.

$$\overline{\tau} = \mu \left[\left(\nabla \vec{v} + \nabla \vec{v}^{\mathrm{T}} \right) - \frac{2}{3} \nabla \cdot \vec{v} I \right] \tag{3.4}$$

For the present analysis, all body forces are neglected and the flow is assumed to be incompressible. The viscosity, μ, is assumed to be constant. These simplifications reduce Eq. 3.3 to the following form.

$$\rho\left(\frac{\partial\vec{v}}{\partial t}+\vec{v}\cdot\nabla\vec{v}\right)=-\nabla p+\mu\nabla^2\vec{v} \tag{3.5}$$

For flows involving heat transfer or compressibility, FLUENT also solves an equation of energy conservation, but for the present analysis of a Micro Air Vehicle the energy equation is not required.

FLUENT solves the governing equations on the mesh using a control volume approach. The computational domain supplied by the user is divided into discrete control volumes and the governing equations are integrated on these individual control volumes. Two different solvers are available: the pressure-based solver and the density-based solver. Since both solvers deal with coupled nonlinear equations, the solution process is iterative.

In the pressure-based solver, the continuity equation is not solved directly, but is used, along with the momentum conservation equation, to derive a pressure-field equation that satisfies continuity [56]. The pressure-based solver has two different algorithms available: the segregated algorithm and the coupled algorithm. In the segregated algorithm, the individual governing equations for flow variables are decoupled and solved sequentially. In the coupled algorithm, the pressure and momentum equations are solved simultaneously, and then governing equations for additional variables are solved sequentially. For both algorithms, iterations are required since the equations are nonlinear and coupled by nature. The segregated algorithm requires less memory than the coupled algorithm, but the convergence is longer for the segregated algorithm because it de-couples the equations.

The density-based approach is generally used for high-speed flows and solves the continuity, momentum, and energy equations simultaneously. Two different formulations are available: implicit and explicit. The implicit approach solves the variables in all cells simultaneously, while the explicit approach solves for each cell individually using a relation based on only known values of variables from the previous iteration or the previous cell calculation. More information on the pressure-based and density-based solvers is available in the FLUENT Theory Guide.

Various options for spatial discretization of the governing equations are available, including the First-Order Upwind scheme, the Power-Law scheme, the Second-Order Upwind scheme, a Central-Differencing scheme, a Bounded Central-Differencing Scheme, a QUICK scheme, a Third-Order MUSCL scheme, and a Modified HRIC

scheme. The order of spatial discretization depends on the scheme used. Detailed information about each scheme and its application is available in the FLUENT Theory Guide.

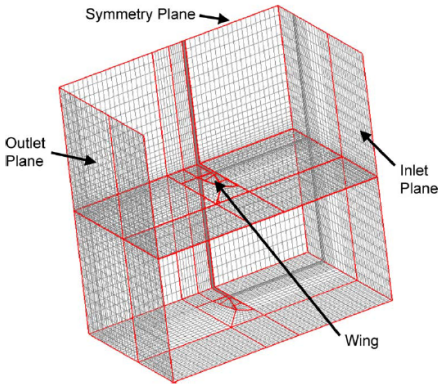
FLUENT has two different options for temporal discretization in unsteady calculations, both of which are first-order: implicit and explicit. The implicit method calculates variables with respect to the future time step and is unconditionally stable with respect to time step size. The explicit method evaluates variables with respect to the current time level, and its stability is dependent on the time-step size: if too large a time step is used, the solution can "blow up" because of numerical error. Explicit time-stepping is only available for the density-based solver. FLUENT also employs various algorithms for evaluating gradients and derivatives and for interpolating pressure and density. Detailed information about these algorithms and their application is available in the FLUENT Theory Guide.

3.1.1 Dynamic Meshing

FLUENT has the capability to model arbitrarily translating and rotating boundaries through its Dynamic Mesh algorithms. Dynamic Meshing deforms and/or remeshes the computational grid in response to changes in the boundaries of the domain. This functionality is normally used in unsteady flows. During the Dynamic Meshing procedure, once the positions of the cells in the computational grid are updated, values of variables are interpolated from the previous grid prior to the solution of the current time step. The Dynamic Meshing algorithm also uses a special version of control volume discretization that allows for movement of the control volume boundary.

FLUENT employs three main Dynamic Meshing procedures: cell smoothing, layering, and remeshing. Cell smoothing may be either spring-based smoothing, by which the computational grid is modeled as a network of springs and node movement throughout the grid is determined by a relation based on Hook's Law, or Laplacian smoothing, which moves mesh vertices to the geometric center of neighboring vertices if mesh quality is improved by such a move. Dynamic layering is used to add or remove layers of cells in hexahedral or wedge mesh zones. Remeshing is a procedure by which individual cells or cell faces are split or merged when cell quality (size or skewness) becomes poor. It is usually used in conjunction with spring-based smoothing. Spring-based smoothing and remeshing in FLUENT are both applicable only for triangular and tetrahedral meshes. More information on Dynamic Meshing is available in the FLUENT Theory Guide.

(a) Experimental flapping actuator



(b) CFD computational domain

Figure 3.1: Methods used by Ho, et al. [17]

3.2 Grid Independence and Model Validation

CFD is a numerical approximation of the governing partial differential equations of fluid flow. The quality of the approximation, and the extendability of its results to real fluid flow problems, is greatly dependent on both the quality of the computational grid used and the various methods used to solve the equations. A great deal of variability in the results of the CFD approximation are possible just by changing the computational grid and solver settings. Because of these issues, any CFD simulation must be subject first to a validation study demonstrating that the results are representative of what would occur in the real flow.

Ideally, a validation study consists of comparisons of the CFD solution against experimental data. For this thesis, at the time the CFD study was performed experimental measurements of the aerodynamic forces on flapping wings from the RIT benchtop simulator were not available, so validation is instead performed against data in the literature. Ho, et al. measured aerodynamic forces on Micro-Air-Vehicle-sized flapping wings experimentally and numerically using CFD in their study on aeroelasticity [17]. Their flapping actuation system allowed for a single degree of freedom, but the inclination of the stroke plane could be varied (see Fig. 3.1(a)—reprinted from Ho, et al. [17]), and simulations were run inside a low-speed wind tunnel.

As a means of initial validation of their CFD model and to perform their own grid independence study, Ho, et al. measured the lift acting on an MAV-sized wing of quarter-ellipse planform operating in a highly-separated flow regime with an angle of attack of 30° and a freestream velocity of 1.7 m/s. The wing was 7 cm in span and 3 cm in chord at the root and had a thickness of 0.01 mm. They also simulated these conditions in CFD using a structured nonuniform grid, which

is shown in Fig. 3.1(b). The computational domain consisted of a large box 42 cm in height, 24 cm in the spanwise direction, and 21 cm in the chordwise direction. These dimensions were chosen to match the dimensions of their wind tunnel. As illustrated in Fig. 3.1(b), the boundaries of the computational domain consisted of an inlet plane, an outlet plane, and a symmetry plane, while the wing was modeled as a no-slip wall. For the grid-independence study, the number of cells ranged from 97,000 to 530,000.

For validation of the FLUENT CFD model, a similar computational domain is constructed. In order to simulate flapping wings in FLUENT, the entire domain may not be structured, because the Dynamic Meshing algorithms employed act on unstructured triangular or tetrahedral cells only. Therefore, a hybrid structured-unstructured grid is used, where the wing is enclosed by an inner volume with a fine structured mesh consisting of prismatic elements, and the outer volume is unstructured and consists of tetrahedral elements. The inner volume does not deform, but instead rotates with the wing to maintain a high-quality mesh near the wing throughout the cycle. Details of the computational model of the wings used for this thesis are given in section 3.3.1. The FLUENT grid-independence study uses grids ranging from 100,000 cells to 600,000 cells. The results of the grid independence study for Ho, et al. and this thesis are presented in Fig. 3.2. Data from Ho, et al. [17] is used with permission. For the grid independence study and all the following studies, the pressure-based coupled solver is used, the QUICK scheme is used for spatial discretization, Green-Gauss Node Based gradient interpolation is used, and standard pressure interpolation is used. For later unsteady flapping simulations, first-order implicit temporal discretization is used. The flow is assumed to be viscous and laminar, which for small flapping MAVs is an assumption well-supported in the literature [6], [23], [34], [45].

Figure 3.2 shows that FLUENT does an excellent job of predicting the lift produced by an MAV wing at high angle of attack, even for low cell counts. Ho, et al. determined that a grid with approximately 500,000 cells would be sufficient to match the experimental result once experimental error is accounted for. Since FLUENT seems to achieve the same level of accuracy on a grid with fewer cells, it is determined here that in FLUENT a grid with roughly 300,000 cells is sufficient.

With the grid-independence study performed, FLUENT is shown to be able to predict the lift on stationary wings at high angles of attack. To further validate the FLUENT solver, flapping simulations are performed to compare against the data of Ho, et al. In their paper, Ho, et al. focused on simulating and measuring the forces on flexible flapping wings, so their CFD solver was validated against experimental

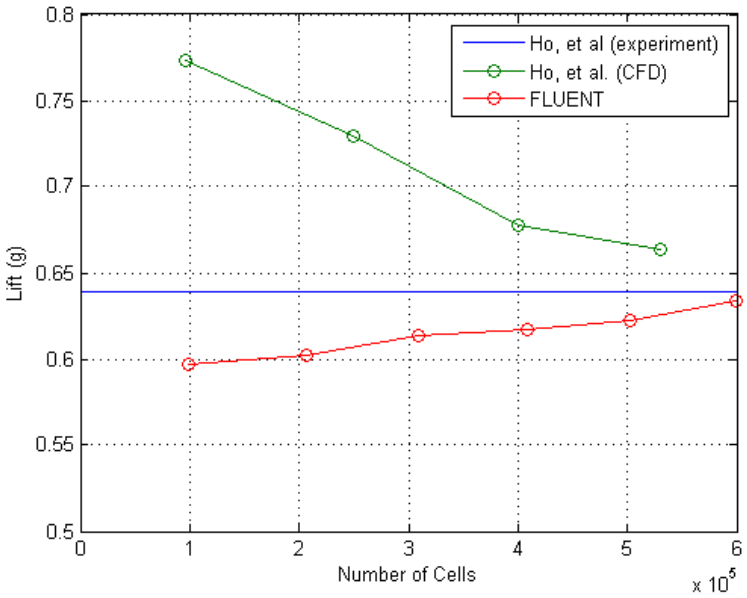


Figure 3.2: Results of grid-independence study

data from flexible wings, using the same computational domain as was used for the grid-independence study. In their validation study, the wings were flapped in a sinusoidal motion on a stroke plane with an inclination of 30° above vertical against an incoming air flow of 1.7 m/s. The total flapping angle (the difference in angle of the wing at each extreme of a stroke, where the wing rotates about a chordwise axis at the root) was 60° and the flapping frequency was 20 Hz. The validation study compared both lift and thrust for the CFD simulation against that measured experimentally, and the results are shown in Fig. 3.3 (reprinted from [17]). The comparison is remarkably close for the two methods, especially considering the fact that their CFD model included aeroelasticity. They attributed the minor differences between the CFD results and the experimental results to the inability to measure the elastic properties of the real wing with absolute certainty.

The FLUENT CFD solver does not have a coupled finite-element module for modeling aeroelasticity, and thus it would not be appropriate to validate FLUENT's results against the results of Ho, et al. that include aeroelasticity. However, Ho, et al. did perform some rigid wing simulations against which the FLUENT solver can be validated. One of these simulations used a rigid rectangular wing, of span 7 cm and chord 3 cm, flapping at 30 Hz in otherwise identical conditions to their previously mentioned validation study. Using the data from this simulation, the FLUENT solver is compared for both lift and thrust over a single flapping cycle. The time step size allows for 90 discrete time steps per cycle, and data is taken from the third full cycle. The results are shown in Fig. 3.4. The data from Ho, et

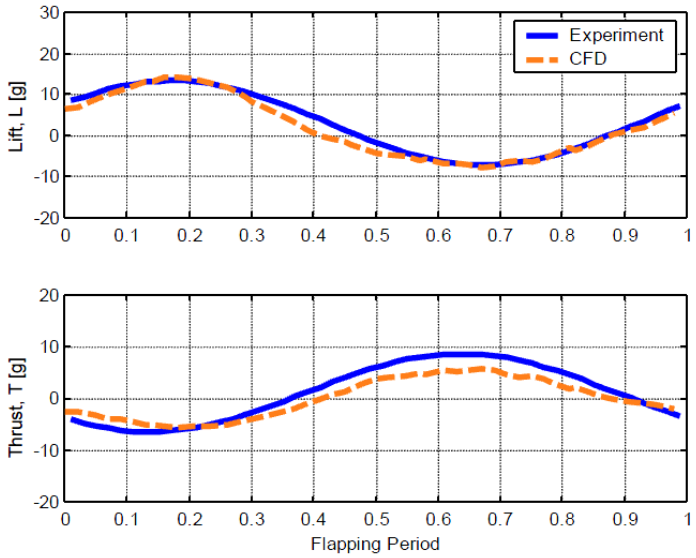


Figure 3.3: Results of CFD validation study by Ho, et al. [17]

al. is used with permission. Though only a single wing is simulated in CFD, the resulting forces are doubled in the figure to account for symmetry in the problem.

As can be seen from Fig. 3.4, FLUENT matches very closely with the data of Ho, et al. The validation study provides confidence that FLUENT can be used to simulate MAV-sized flapping wings in highly unsteady and separated flows. The grid-independence and validation studies also provide guidance on how large of a computational grid to use in later simulations and which solvers and methods in FLUENT work best for the simulation of MAV flapping flight.

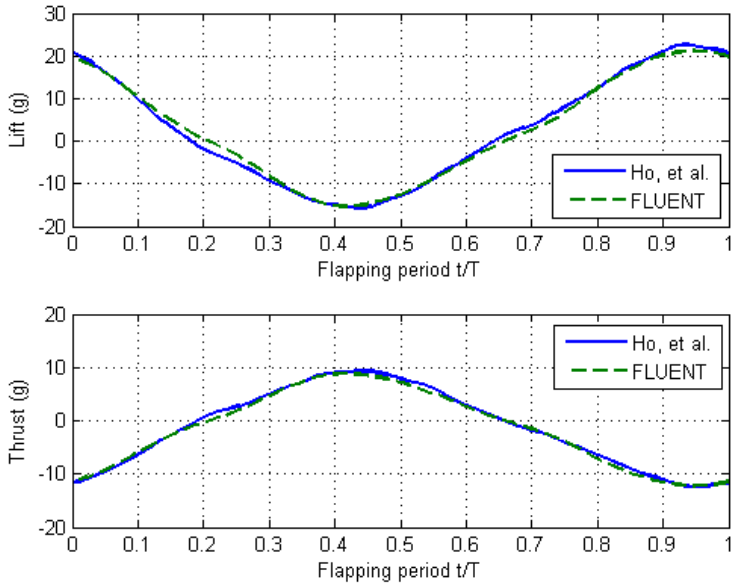


Figure 3.4: Results of FLUENT validation study

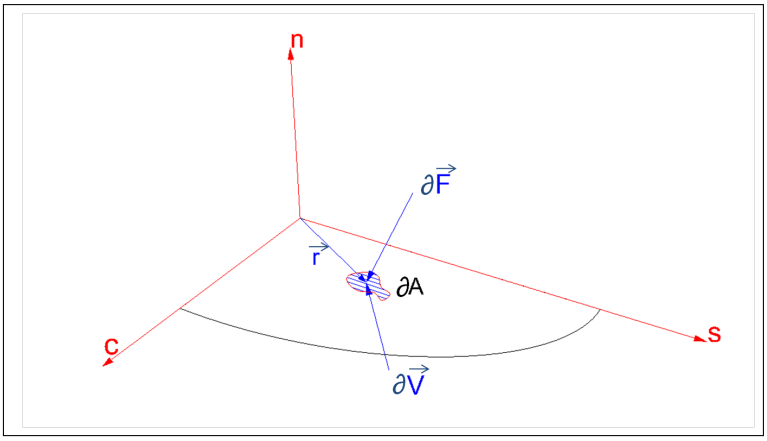


Figure 3.5: Derivation of aerodynamic power requirement

3.3 Single-Wing Parameter Sweeps

The major goals of the CFD analysis of flapping wings for the QV are to gain insight into the vertical force that can be produced by the flapping wings in hovering mode, as well as the power input required, and to determine the flapping frequency, amplitude, and feathering characteristics of the wing's motion that will result in the best performance. For the analysis, the optimal kinematics parameters are the ones that allow the QV to achieve enough vertical force to hover while minimizing the power requirement.

The power requirement for hovering is comprised of two major components: power required to overcome aerodynamic moments and power required to overcome inertial torques. The QV design implements elastic energy storage during flapping to reduce the inertial power requirement. If the spring system is tuned to the natural frequency of the flapping system, the inertial power requirement is provided by the passive spring system instead of the actuators [2], [19]. Thus, the aerodynamic power requirement mainly determines whether a design is "good" or not.

The aerodynamic power required by a flapping wing is determined using first principles. Consider a differential area ∂A of a flapping wing at a location $\vec{r}$ relative to the origin, which is the center of wing rotation, as shown in Fig. 3.5. The wing is rotating such that this differential area has a kinematic velocity $\vec{V}$ and experiences a differential aerodynamic force $\partial\vec{F}$. The total force acting on the wing is obtained by integrating $\partial\vec{F}$ over the entire wing area. The differential power input to move this area ∂A against the aerodynamic force is given by Eq. 3.6.

$$\partial P = -\partial\vec{F} \cdot \vec{V} \tag{3.6}$$

The negative sign in the equation above is there because we are interested in the

power put *into* the system by an external source. For a rotating wing, the kinematic velocity of an arbitrary point is

$$\vec{V} = \vec{\Omega} \times \vec{r} \tag{3.7}$$

where $\vec{\Omega}$ is the rotational velocity. Substituting this relation into Eq. 3.6, the following relation is returned.

$$\partial P = -\partial \vec{F} \cdot \left(\vec{\Omega} \times \vec{r} \right) \tag{3.8}$$

Using a property of the dot product, Eq. 3.8 can be written in the following form:

$$\partial P = -\vec{\Omega} \cdot \left(\vec{r} \times \partial \vec{F} \right) = -\vec{\Omega} \cdot \partial \vec{M} \tag{3.9}$$

where $\vec{M}$ is the aerodynamic moment. For a rigid wing, the rotational velocity $\vec{\Omega}$ is constant across the entire area, so this equation can be integrated over the entire wing to result in the final definition of aerodynamic power requirement, given in Eq. 3.10.

$$P = -\vec{\Omega} \cdot \vec{M} \tag{3.10}$$

Since the CFD method of analysis readily reports the total aerodynamic moment acting on the wing at any given time, and since the rotational velocity is prescribed, this definition of aerodynamic power requirement is a convenient one to use. Equation 3.10 is also used by Azuma, et al. [19] and by Liu and Sun [40].

The proposed Quad-wing Vehicle (QV) is designed with four wings in a dragonfly-like configuration and a near-vertical stroke plane. In the downstroke the relative angle of attack is very high to produce vertical force primarily via drag. Prior to the upstroke, the wing rotates or *feathers* so that the relative angle of attack is small and a small amount of downward force is produced. Depending on the feathering angle, the wing can also produce a net horizontal force during the upstroke. In the QV, feathering will likely be achieved passively using springs. The wing undergoes two different rotations during a flapping cycle: a flapping rotation about the body x-axis and a feathering rotation about the wing's spar axis as shown in Fig. 3.6. In this figure, the s-axis is the wing's leading-edge axis, the c-axis is the wing's root-chord axis, and the x-, y- and z-axes are the MAV body axes, which for hovering flight are also inertial axes. For the CFD study, the wing rotational velocities are assumed to be simple sinusoids and are given by the following equations.

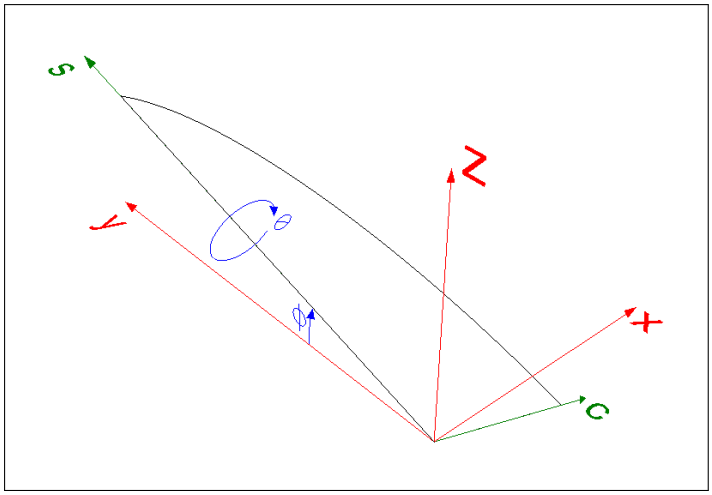


Figure 3.6: QV wing rotations

$$\dot{\phi} = -2\pi\phi_{max} f \cos(\omega t) \tag{3.11}$$

$$\dot{\theta} = \frac{2\pi\theta_{max} f \sin(\omega t)}{2} \tag{3.12}$$

In the CFD analysis, the three major flapping kinematics parameters are the flapping frequency f, the flapping amplitude ϕ_{max} (this is half of the *total* flapping angle identified by Ho, et al.), and the total feathering angle θ_{max}, which is a rotation of the wing about its own leading edge. For the parameter sweeps, f is varied from 5 Hz to 35 Hz in increments of 5 Hz, ϕ_{max} is varied from 20° to 90° in increments of 10°, and θ_{max} is varied from 20° to 90° in increments of 10°.

Given these ranges of kinematics parameters to be swept, systematically investigating the entire design space would require 448 total CFD simulations, which is far too many to be tractable given the time requirement to perform a single simulation and to process the data. Instead, 56 initial simulations are performed, sweeping f and ϕ_{max} while holding θ_{max} constant at a moderate value of 60°. This approach is taken because f and ϕ_{max} are considered to be the two most important of the flapping parameters, as they determine the flapping velocity of the wing during the stroke. After analyzing the results of the first 56 simulations, f and ϕ_{max} are held constant at their "optimum" values to determine the effect of feathering amplitude on the forces produced and the power required. The same thing is also done for a couple other representative values of f and ϕ_{max} to determine whether the effect of feathering amplitude holds true in general.

The layout of the QV constrains the stroke plane to be vertical or near-vertical. In the CFD simulations, the wings are assumed to flap in a vertical stroke plane.

The wing is horizontal in the middle of the downstroke and near vertical during the middle of the upstroke. This motion results in an angle of attack near 90° for most of the downstroke but it is much smaller for most of the upstroke, as illustrated in Fig. 1.10. Thus, the wing produces vertical force to aid in hovering primarily via drag during the downstroke.

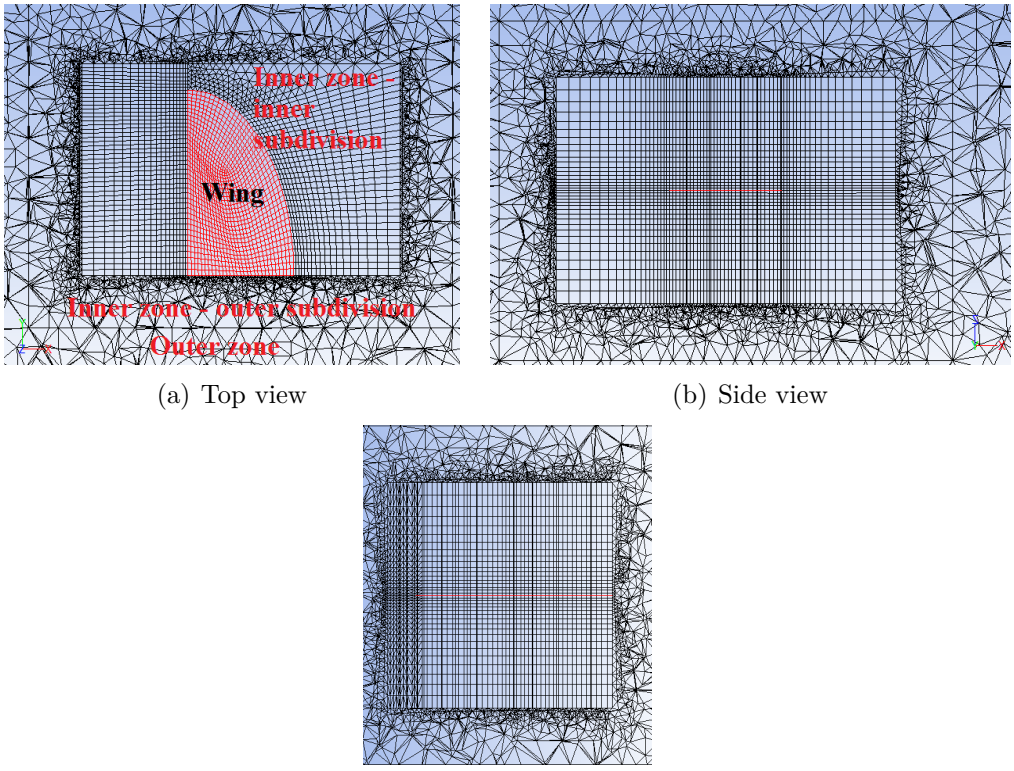
Though the CFD simulations use a completely vertical stroke plane, it is worth noting that during the design stage the inclination of this stroke plane may be changed slightly to achieve the greatest possible net vertical force and the minimum possible net horizontal force. Because of this, the *usable hovering force* is defined not simply as the vertical force from CFD, but the vector sum of the vertical force (z-direction in CFD) and the horizontal force in the forward-and-backward direction (x-direction in CFD). Designs are evaluated based on this usable hovering force, or F_h, which is given in Eq. 3.13.

$$F_h = \sqrt{F_x^2 + F_z^2} \tag{3.13}$$

This equation is used with the assumption that the angle of inclination of the stroke plane away from vertical, as required to make use of F_h for hovering, is small so as not to require a change in the layout of the QV.

3.3.1 Computational Model and Methods

For the single-wing parameter sweeps, it is assumed that interference effects between the wings are minimal and that the contribution of the body to the aerodynamic forces is also minimal. Thus, only a single wing is modeled. Additionally, body oscillations due to time-varying forces from the flapping wings are neglected. The computational model of the wing is constructed in GAMBIT. The wing geometry is similar to that used in the grid independence study. It is assumed to be a flat quarter-ellipse planform with a span of 7 cm and a root chord length of 4 cm. The span of 7 cm is selected because it is roughly the max span possible for a single MAV wing in order to keep the total wingspan at 15 cm or lower when the body and the other wing are accounted for. Though real wings have some thickness, for flapping MAV and insect wings the thickness is usually very small to keep weight low. The computational model is assumed to be a flat plate with no thickness. Simulating the wing as a flat plate has benefits when creating the mesh, because typically for a mesh around a very thin body the computational cells interfacing with the thin face of the body have a high aspect ratio. For a wing with no thickness, the thin face of the wing is reduced to a single edge and those high-aspect-ratio cells are



(a) Top view

(b) Side view

(c) Front view

Figure 3.7: Mesh on and around the wing

removed. The mesh immediately surrounding the wing is shown in Fig. 3.7.

As mentioned in section 3.2, the computational domain is divided up into two separate zones: an outer zone consisting of unstructured tetrahedral elements that deform and are remeshed to allow for movement of the inner zone, and an inner zone which rotates with the wing and is subdivided into two zones as well. The innermost of the subdivisions immediately surrounds the wing and is made up of structured prismatic elements. The outer subdivision is made up of unstructured tetrahedral elements and fulfills the purpose of smoothly interfacing the inner zone with the outer zone. This is necessary when using the Dynamic Mesh in FLUENT, because FLUENT may not perform spring-based smoothing or remeshing operations on the wedge cells that result at the interface between a structured zone and an unstructured zone. The two major zones and subdivisions are highlighted in Fig. 3.7(a).

The internal boundaries between zones are only computational boundaries; they do not create any obstruction for the flow in the CFD simulation. There are three benefits to this multi-zone method. The first is that it ensures the mesh near the wing remains of constant high quality throughout the simulation, improving the accuracy. The second is that, though the linear velocities of the boundaries of the

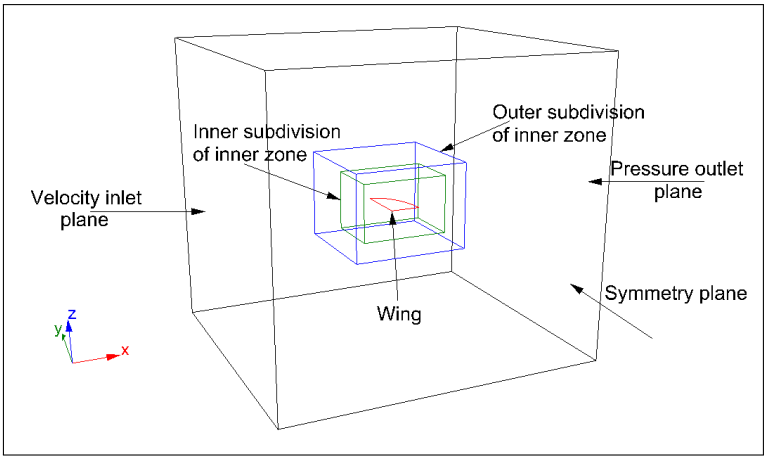


Figure 3.8: CFD domain used for single-wing simulations

inner zone are higher than those of the wing itself, the size of the deforming cells in the outer zone may also be larger than those in the immediate vicinity of the wing. This allows a larger time step to be used without creating negative cell volumes. The third benefit is that there are fewer cells that must be deformed and remeshed during each time step, saving a considerable amount of computational time. The major downside of this method is that additional space must be allotted within the domain to ensure the boundaries of the inner zone do not collide with any stationary boundaries. This means more space must be allowed between the wing and the symmetry plane. In accordance with the results of the grid independence study, the mesh consists of roughly 320,000 individual cells, most of which are contained in the inner zone.

The boundary conditions for the computational domain are the same as for the grid independence study. The outer box is formed by a velocity inlet plane, a pressure outlet plane, three side walls, and a symmetry plane. For hovering simulations, the velocity at the inlet and the gauge pressure at the outlet are both set to 0. The wing is modeled as a non-slip wall. In FLUENT, a flat plate surface, such as a wall, is computationally considered to have only one side. Thus, FLUENT automatically assigns a coupled mirror wing surface whose area vector faces in the opposite direction of the original wing and has the same boundary condition.

The outer box has dimensions 46 cm in the x-direction, 41 cm in the y-direction, and 42 cm in the z-direction. The wing is centered in the domain in the vertical and chordwise directions. The inner box is 16 cm in the chordwise direction, 15 cm in the spanwise direction, and 12 cm in the normal direction, and the wing is also centered in the inner box in the vertical and chordwise direction. The root of the wing is 13 cm from the symmetry plane to allow room for the inner zone to rotate. The innermost subdivision in the domain, which contains the wing and structured

cells, is 8 cm in the spanwise direction, 12 cm in the chordwise direction, and 8 cm in the normal direction. In each simulation, the wing starts in the middle of the downstroke with ϕ and θ equal to 0, so that the chordwise direction is aligned with the x-direction, the spanwise direction is aligned with the y-direction, and the normal direction is aligned with the z-direction. The computational domain is illustrated in Fig. 3.8.

For the simulations, the same solver settings were used as in the validation study. The solver is the pressure-based coupled algorithm and assumes unsteady viscous laminar flow. Spatial discretization is the QUICK scheme, which is a higher-order discretization for quadrilateral and hexahedral elements based on a weighted average of second-order upwind and central difference interpolations [56]. For tetrahedral cells, the second-order upwind scheme is used instead. Green-Gauss Node Based gradient interpolation and standard pressure interpolation are used. The Green-Gauss Node Based interpolation is used because it is recommended for unstructured meshes in the FLUENT Theory Guide [56]. Standard pressure interpolation is used because it gave good results in the validation study and because the alternative schemes did not appreciably alter the predicted forces. Time advancement is first-order implicit.

Dynamic Mesh settings are important when using the Dynamic Mesh functionality in FLUENT. For these simulations, spring-based smoothing and local remeshing are used. The spring-based smoothing spring constant factor is set to a very low value of 0.001. This causes the cells farther away from the moving zone, where flow gradients are generally lower, to deform more so than the cells near the moving zone, which instead tend to move along with the moving zone. Thus, a low spring constant factor helps to preserve the quality of the mesh near the moving zone. For local remeshing, several criteria are set which determine when cells will be marked for remeshing. These criteria are minimum and maximum length scale and maximum cell skewness. Any cells of length scale smaller than the minimum or greater than the maximum size are marked for remeshing, and any cells with skewness greater than the maximum skewness are also marked for remeshing. For these simulations, the minimum length scale is set to the minimum cell edge length in the deforming zone for the initial mesh, which is 1 cm. The maximum length scale is set to ten times the minimum, and the maximum skewness is set to the default value of 0.7.

The time step for each simulation follows from guidelines established during the validation study. In the validation study, the flapping amplitude ϕ_{max} was 30° and the time step allowed for 90 time steps per cycle. In the parameter sweeps, the

flapping amplitude may be as great as 90°. If the flapping cycle is divided into a certain number of time steps, n_{steps}, which is independent of the flapping frequency, the circumferential distance traveled by the wing tip in each time step is a function of ϕ_{max} and the number of time steps per cycle only. For larger values of ϕ_{max}, the circumferential distance traveled per time step is increased if the number of time steps is held constant. This can potentially result in negative cell volumes when the wing moves from one time step to the next and can also decrease the accuracy of the results. Because of this, the number of time steps is set dependent on ϕ_{max} according to the relation given in Eq. 3.14,

$$n_{steps} = 30\phi_{max} \tag{3.14}$$

where ϕ_{max} is given in degrees. The exception to this relation is when ϕ_{max} is equal to 20°, for which n_{steps} is set to 90 instead of 60. This is so that n_{steps} is never any smaller than in the validation study. For any simulations where ϕ_{max} is not a multiple of 10, n_{steps} is rounded up to the nearest 10. The physical time step is related to n_{steps} by the flapping frequency, as given in Eq. 3.15.

$$\Delta t = \frac{1}{f \cdot n_{steps}} \tag{3.15}$$

For each simulation, three full flapping cycles are run, and data is taken from the third cycle. This approach is in accordance with the validation study. For the longest simulations, when ϕ_{max} is 90°, 630 total time steps are needed. Each of these longest simulations requires roughly 19 hours on a desktop computer with an Intel Xeon 2.67 GHz quad-core processor and 4 GB RAM. The computing time is directly proportional to the total number of time steps required for each simulation.

During the solution process, the FLUENT solver undergoes iterations in addition to time steps. Each time step requires a certain number of iterations in order to drive the residues of the governing equations to a minimum. For the flapping-wing MAV simulations with the pressure-based coupled solver, it was found that 40 iterations is sufficient for the residues to reach a minimum. A typical iteration-history of residues is shown in Fig. 3.9. In this figure, the first 6 time steps and part of the seventh are shown. The residue of the continuity equation reaches approximately 10^{-4}, while each component of the momentum equation reaches approximately 10^{-7}. The reason why the velocity residues are lower than the continuity residue is not known for certain, but it was the case for all of the 3D simulations presented in this chapter as well as the 2D simulations in Chapter 4. Note that the convergence improves after the first couple time steps.

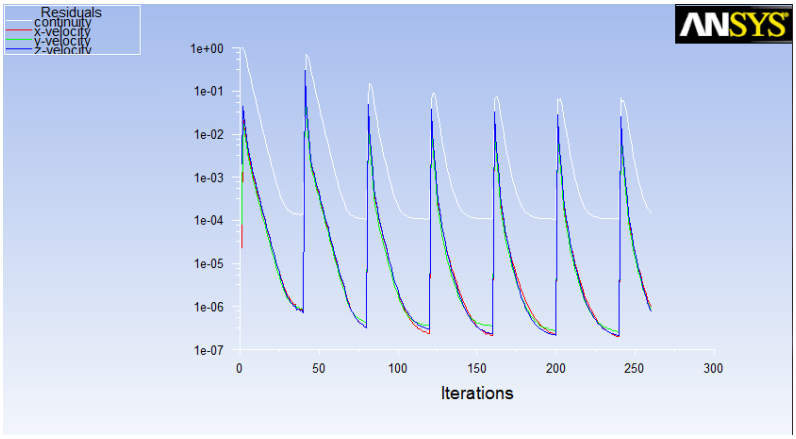


Figure 3.9: Typical iteration-history of residuals

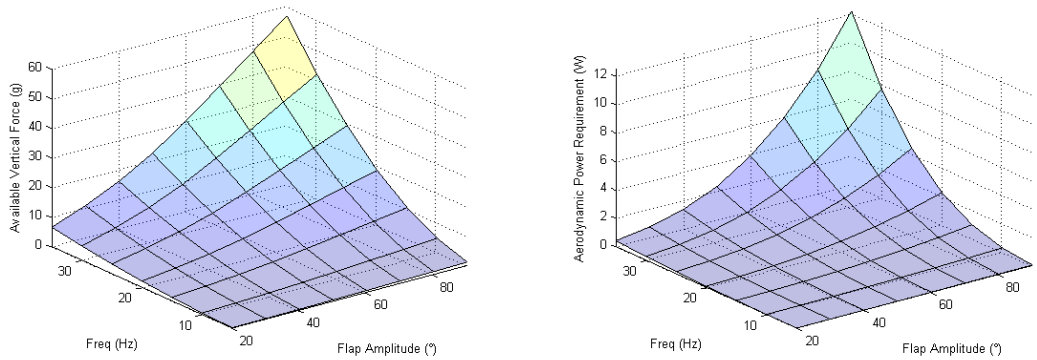
3.3.2 Results

Aerodynamic forces and power requirement

Initially, 56 total simulations are performed, varying ϕ_{max} and f while holding θ_{max} constant at 60°. At the end of each time step, FLUENT calculates and reports the three components of force and the three components of moment acting on the wing. The power requirement is calculated using Eq. 3.10. $\vec{\Omega}$ is prescribed, and its x-, y-, and z-components are related to the flapping angle ϕ and the feathering angle θ by Eq. 3.16. Note that the flapping rotation, ϕ, occurs about the inertial x-axis, but the feathering rotation, θ, occurs about the leading-edge s-axis, as shown in Fig. 3.6. Because the leading-edge moves with the wing, the feathering rotation is manifested in both the y- and z-components of $\vec{\Omega}$, but the flapping rotation is only in the x-component. Initially, the wing is in the middle of the downstroke and is flat, so ϕ and θ are both 0. $\dot{\phi}$ and $\dot{\theta}$ are given by Eqs. 3.11 and 3.12. An illustration of the kinematics is given in Fig. 1.10.

$$\vec{\Omega} = \left\{ \begin{array}{c} \dot{\phi} \\ \dot{\theta}\cos\phi \\ \dot{\theta}\sin\phi \end{array} \right\} \tag{3.16}$$

For each simulation, cycle-averaged forces and power requirement from the third flapping cycle are used to construct surface plots of the F_h and P design spaces. The cycle-averaged forces are used because they represent the net effect on the MAV for a flapping cycle. In reality the body will oscillate during the cycle due to the variation of the forces with time, but this effect was shown to generally be of minor importance by Wu, et al. [44]. The usable hovering force and power requirement are shown in Fig. 3.10. In these surface plots, though only a single wing is simulated



(a) Usable hovering force (g)

(b) Power (W)

Figure 3.10: Usable hovering force and aerodynamic power requirement

in CFD, the forces and power requirement are multiplied by a factor of 4 to give a better idea of what can be expected from the Quad-Wing Vehicle. In this figure, F_h is given in grams force and P is given in watts. These surface plots represent flapping with a maximum feathering angle of 60°, which occurs during the middle of the upstroke. The data used to create these surface plots is given in Appendix A.

Not surprisingly, both the usable hovering force and aerodynamic power requirement increase quickly as ϕ_{max} and f are increased. Neither of the surface plots represents a linear relationship between the design variables and the output quantities, but F_h seems to increase more linearly than P as the values of the design variables increase. Since the hovering force required for the MAV is equal to its weight, this result shows that great savings in the power requirement can be realized by decreasing the weight of the MAV by even minor amounts. At the highest values of ϕ_{max} and f, the QV produces 57 g of hovering force and requires 12 W to overcome the aerodynamic moments.

These surface plots are useful in showing the F_h and P landscapes, but they do not provide clear direction on how to optimize the design by themselves. It is also necessary to set an F_h constraint and, using that constraint, to pick a set of parameters that meet it while minimizing the power required. According to early estimates from the group at Georgia Tech, the QV will require about 35 g of hovering force to lift its own weight and payload of sensors. Fortunately, the CFD simulations show that this is within the realm of possibility, provided the actuation system is able to beat the wings at the necessary frequency and flapping angle. As a visual tool to help optimize the design, F_h isocurves are computed in MATLAB using a linear interpolation method and are plotted on the P surface as is shown in Fig. 3.11. The constraint of 35 g is highlighted.

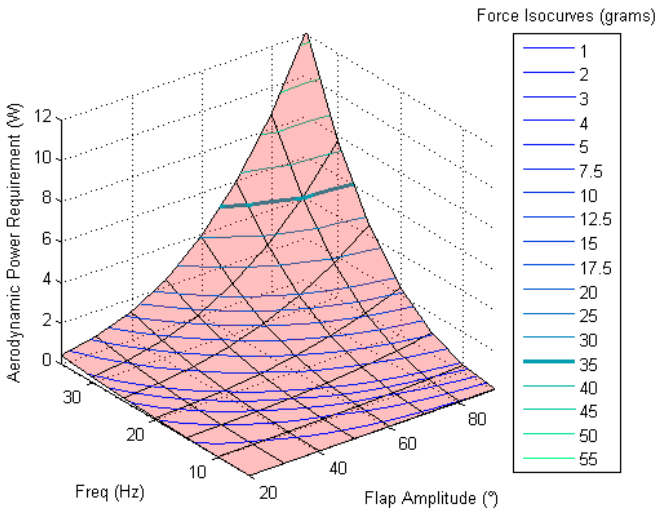


Figure 3.11: Power requirement with F_h isocurves

Using this figure, it is apparent that the range of ϕ_{max} and f able to generate 35 g of usable hovering force is fairly small. On one end of the curve, the flapping frequency is 35 Hz and the flapping amplitude is 65.4°; on the other end, the flapping frequency is 27.4 Hz and the flapping amplitude is 90°. Either way, the actuation system will need to achieve a quite high frequency and amplitude. Fortunately, the aerodynamic power input required for this level is fairly low. The CFD analysis predicts that, at the former end of the 35 g isocurve where f is 35 Hz, 5.0 W is required to overcome the aerodynamic moments on all four wings. At the other end of the isocurve where ϕ_{max} is 90°, the aerodynamic power requirement is increased to 6.0 W. Bear in mind also that these numbers are with a feathering amplitude of 60°; better performance can be realized once that parameter is optimized as well.

Since for the 35 g isocurve the power requirement increases with increasing ϕ_{max}, for minimum power requirement the frequency should instead be pushed to the maximum limit allowed by the actuation hardware. However, this local result should not necessarily be generally followed as a design guideline, because in future iterations of the design the force requirement may change. Figure 3.12 shows a side view of the power surface with ϕ_{max} along the horizontal axis. Using this figure, it is clear that the local result described for the 35 g isocurve holds true in general, at least when θ_{max} is 60°: for the minimum aerodynamic power requirement the flapping frequency should be increased to the max instead of increasing the flapping amplitude. The effect increases as F_h increases, whereas for very low F_h the isocurves are fairly flat in the P direction. In addition to the power savings, another benefit of driving the flapping frequency, instead of the flapping amplitude, to the max is that it allows additional amplitude for maneuvering, which is important for

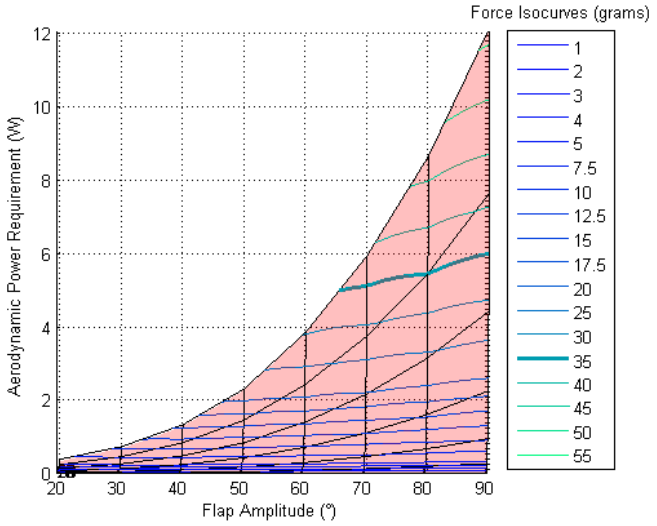


Figure 3.12: F_h isocurves side view

the QV since it is designed to fly with a constant flapping frequency.

Following the initial runs with θ_{max} held constant, since the best design meeting the lift constraint, which had ϕ_{max} equal to 65.4° and f equal to 35 Hz, was determined by interpolating the data, a simulation was then performed using those values of the design variables. For this case, the net usable force is made up of a 31.5-g-component in the z-direction and a 15.2-g-component in the negative x-direction. Horizontal force is produced mostly in the upstroke, for most of which the wing's angle of attack is about 30°. These force components dictate that the stroke plane needs to be rotated 25.8° from vertical in order to produce only a net vertical force. This is a significant change in the stroke plane inclination, but not so much as to cause problems with the forewings and hindwings contacting each other in the QV.

The time histories of F_z, F_x, and F_h are shown in Fig. 3.14(a). The time history for F_h is calculated from F_z, F_x, and the stroke plane inclination δ using Eq. 3.17. The concept of rotating the stroke plane is illustrated in Fig. 3.13.

$$F_h = F_z \cos\delta - F_x \sin\delta \tag{3.17}$$

In Fig. 3.14(a) bear in mind that the downstroke spans from t/T of 2 to 2.25 and 2.75 to 3, while the upstroke occurs between t/T of 2.25 and 2.75. The maximum feathering rotation speed occurs at t/T of 2.25 and 2.75. The vertical force, F_z, in the simulation is large during the downstroke and reaches a peak just after the middle of the stroke. During the upstroke, F_z becomes negative due to the angle of attack of the wing. The horizontal force during the upstroke is in the negative

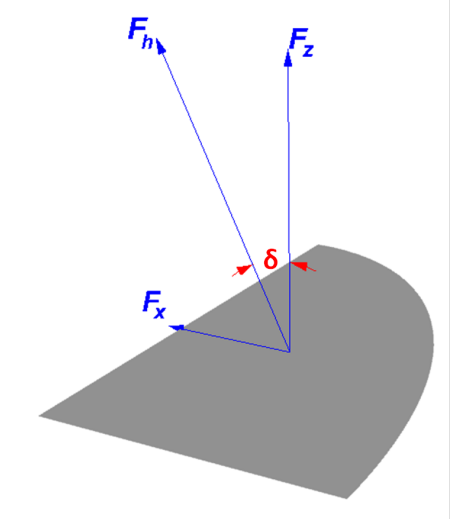
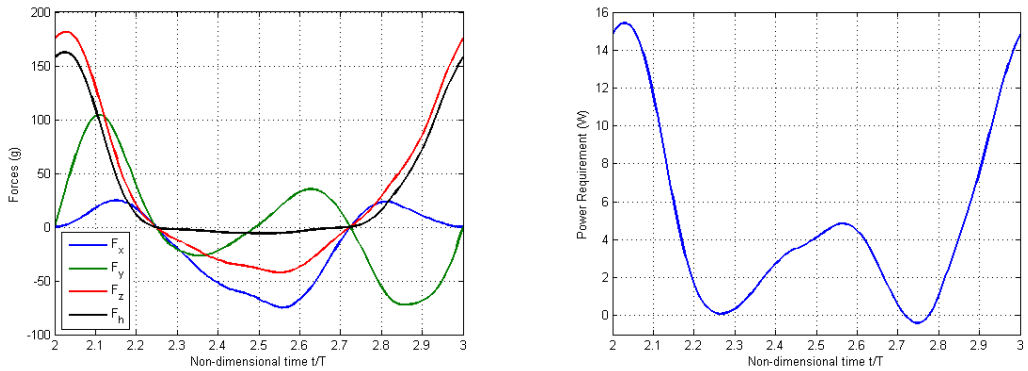


Figure 3.13: Stroke plane inclination angle



(a) Forces (g)

(b) Power requirement (W)

Figure 3.14: Forces and power requirement for the initial best design

x-direction, which for the QV would be seen as thrust, though in reality it is due to lift force on the wing when the stroke plane is vertical. The side force, F_y, varies throughout the cycle. The time history of F_h shows that when the stroke plane is inclined by the angle δ, the cycle-averaged usable hovering force is increased primarily by decreasing the negative force during the upstroke.

The aerodynamic power requirement, shown in Fig. 3.14(b), also fluctuates throughout the cycle, reaching a peak at the same point that F_z and F_h do. The power required is much higher during the downstroke than during the upstroke because of the very high angle of attack during the downstroke. This suggests that using drag for hovering force may not be the most energy-efficient option. Somewhat interesting to note is that for a brief time during pronation (t/T = 2.75), the power requirement is actually slightly negative, showing that the aerodynamic moments are acting in the same direction as the feathering rotation at that point. During

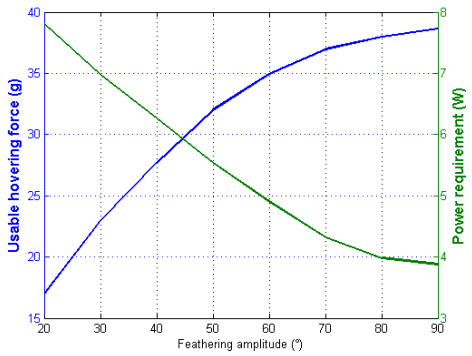


Figure 3.15: Dependence of F_h and P on θ_{max}

supination, the power requirement approaches zero but does not become negative.

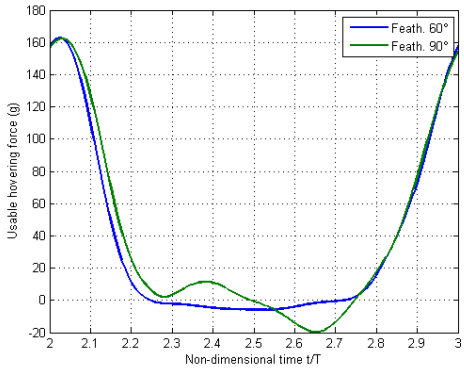
Effect of changing feathering amplitude

Following the initial 56 simulations holding θ_{max} constant at 60°, θ_{max} is then swept from 20° to 90° in increments of 10°. The results of the initial parameter sweeps show that the wing produces a net horizontal force during the upstroke, which, if the stroke plane is inclined by the angle δ from vertical, must be balanced during the downstroke. For hovering flight this unneeded horizontal force contributes to the power requirement without providing vertical force for hovering. Thus, it seems reasonable that increasing the feathering amplitude θ_{max} should decrease the overall power requirement. Increasing the feathering amplitude should also decrease the required stroke plane inclination, δ, because there will be less horizontal force produced during the upstroke. Since the stroke plane inclination also decreases the vertical force produced during the downstroke, it is likely that increasing the feathering amplitude would increase the cycle-averaged vertical force as well.

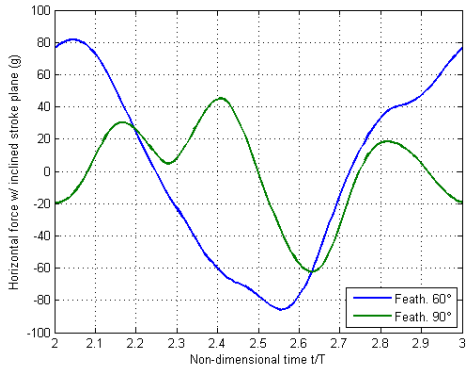
Figure 3.15 plots F_h and P versus feathering amplitude for the case where ϕ_{max} is 65.4° and f is 35 Hz. This figure shows that increasing the feathering amplitude does indeed decrease the aerodynamic power requirement while simultaneously increasing the usable hovering force. In this simulation, increasing the feathering amplitude from 60° to 90° increases F_h from 35 g to 39 g and decreases the power requirement from 5 W to 3.9 W. Both the power requirement and usable hovering force seem to level off near θ_{max} of 90°. For θ_{max} of 90°, with a vertical stroke plane a net horizontal force is produced in the positive x-direction, which means the required stroke plane inclination to make use of F_h is negative. In this case, the F_z component is 38.4 g in the positive z-direction and the F_x component is 4.8 g in the positive x-direction, so δ is $-7.1°$.

Figure 3.16 compares the F_h and P for the case with θ_{max} equal to 90° against

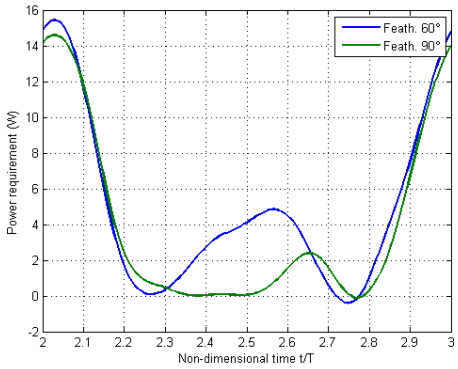
the case with θ_{max} equal to 60° (forces and power for this case are also shown in Fig. 3.14). Figure 3.16(b) shows the horizontal forces for each case when the appropriate stroke plane inclination is applied. For both cases, the horizontal force varies throughout the cycle, with a cycle-average of 0, but for the higher feathering amplitude the magnitude of the force in each direction is smaller. Figure 3.16(a) shows that the vertical force is in general greater when feathering amplitude is increased to 90°, both in the downstroke and for much of the upstroke. F_h does become negative for some of the upstroke with θ_{max} equal to 90°, likely due to the effect of feathering rotation. Figure 3.16(c) shows that aerodynamic power is also decreased for the higher feathering amplitude, especially during the upstroke.



(a) F_h (g)



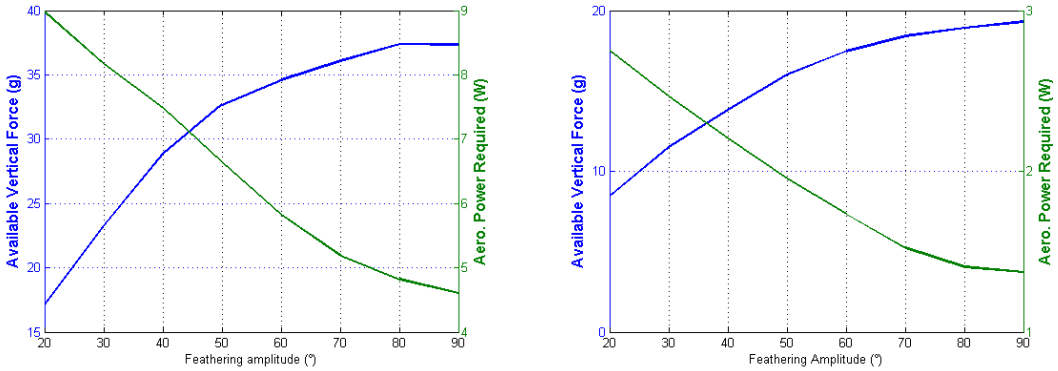
(b) Horizontal force (g)



(c) Power (W)

Figure 3.16: Force and power traces for different feathering amplitudes

In order to determine whether these trends regarding the feathering amplitude are true in general, θ_{max} is swept for a couple other values of ϕ_{max} and f. The first set is on the other side of the 35-g-F_h isocurve from Fig. 3.11, where ϕ_{max} is 90° and f is 27.4 Hz. The second set is in the middle of the 17.5-g-F_h isocurve and has ϕ_{max} equal to 65° and f equal to 24.9 Hz. The dependence of F_h and P on θ_{max} is shown in Fig. 3.17. The same trends are apparent here as were seen in the previous case: increasing the feathering amplitude towards 90° increases F_h and decreases



(a) $\phi_{max} = 90°$, $f = 27.4$ Hz (b) $\phi_{max} = 65°$, $f = 24.9$ Hz

Figure 3.17: Effect of feathering amplitude at other design points

P. Thus, in general, when flapping wings are designed to produce hovering force primarily via drag, the best performance is attained when the feathering amplitude is around 90°. Another interesting thing to note is that in Fig. 3.17(a), the power curve is higher than in Fig. 3.15 and the hovering force curve is equal or lower. This supports the finding that a high flapping frequency, instead of a high flapping amplitude, should be used to maximize usable hovering force.

Effect of scaling

The kinematics parameter sweeps described in the previous section were performed for a wing sized 7 cm in span and 4 cm in chord. For the actual QV, however, these exact sizes may not be used. The aspect ratio, for one thing, likely has some effect on the forces and power required. However, the area of the wings is a bigger factor, because the forces produced by any wing are directly related to the area. The formula for aerodynamic force on a wing in steady flow is given by Eq. 3.18,

$$F = \frac{1}{2} C_F \rho V^2 S \tag{3.18}$$

where V is the upstream air velocity, S is the planform area, ρ is the air density, and C_F is the coefficient of force. In hovering flight, since the wing is undergoing rotational motion, the characteristic velocity seen by the wing is a function of the characteristic length d from the center of rotation (for example, the distance from root to tip) and the rotational velocity. If the wing kinematics are held constant for two different-size wings, then the characteristic velocity is a function of d only; thus, V^2 is proportional to d^2. If the size of the wing is to be changed but the shape is to be held constant, i.e. if the whole QV were to be scaled down, d^2 is directly proportional to S. Considering Eq. 3.18, one would then expect the forces to be

directly proportional to S^2 as in Eq. 3.19.

$$F \propto S^2 \tag{3.19}$$

The power requirement should also be related to the wing area. Recall from Eq. 3.10 that the aerodynamic power requirement is a function of the rotational velocity and the aerodynamic moments on the wing. The formula for aerodynamic moment acting on a wing in steady flight is given by Eq. 3.20,

$$M = \frac{1}{2} C_M \rho V^2 S d \tag{3.20}$$

where C_M is the moment coefficient. Following the same logic as was used to determine the dependence of the forces on the wing area, one would expect the power requirement to be proportional to $S^{5/2}$, as in Eq. 3.21.

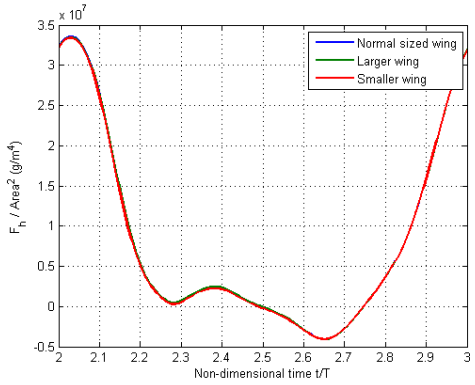
$$P \propto S^{5/2} \tag{3.21}$$

To test the validity of these relations, the wing in the previous simulations was scaled twice to create one wing with double the original area and another wing with half the original area. The kinematics parameters are $\phi_{max} = 65.4°$, $f = 35$ Hz, and $\theta_{max} = 90°$. The relationships in Eqs. 3.19 and 3.21 are tested by plotting F_h/S^2 and $P/S^{5/2}$ for the three cases. The results are shown in Fig. 3.18. The comparisons show that the dependence of forces and power on area developed here holds true. These relationships will be important if the final QV design is to be scaled differently than the computational simulations performed in this thesis. The agreement among the three cases is remarkably close, showing that the aerodynamic phenomena acting on the MAV wings are not greatly affected by reasonable changes in scale.

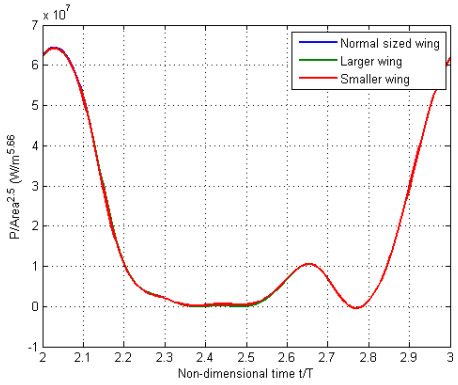
Flow field characteristics

For flow field visualizations, the "optimal" case with ϕ_{max} equal to 65.4°, f equal to 35 Hz, and θ_{max} equal to 90°, with span equal to 7 cm and root chord equal to 4 cm, is used to create flow visualizations.

Figure 3.19 shows the contours of static pressure on the top surface of the wing during the complete cycle (note that the viewing angle changes after the first three images to better see the top surface). At the beginning of the downstroke the wing is undergoing purely rotational motion, which causes high pressure on the top surface of the wing. For the rest of the downstroke the top surface of the wing experiences



(a) Scaled usable hovering force



(b) Scaled aerodynamic power requirement

Figure 3.18: Effect of wing scale on forces and power requirement

low pressure, creating vertical force via drag. After the middle of the downstroke, the low pressure begins to be localized into regions near the leading edge, trailing edge, and root, while higher pressure creeps into the other regions. During the first half of the upstroke, though the angle of attack is very low, the feathering rotational velocity is acting to create a net upwash on the wing, decreasing the pressure on the top surface and producing a positive vertical force. During the second half of the upstroke, the opposite effect can be seen, as the pressure on the top surface becomes high and negative vertical force is produced. The effect of rotational motion during the upstroke can also be seen in the force trace in Fig. 3.16(a).

There are many references in the literature to the leading-edge vortex acting on the wings of insects in hovering mode. Most of the studies on leading-edge vortices deal with wings that are flapping in a horizontal or nearly horizontal stroke plane. In such cases, the leading-edge vortex allows for high amounts of lift to be produced by a flapping wing at high angle of attack because it creates a suction zone which allows the flow to reattach on the top surface of the wing. For the QV, the stroke plane is nearly vertical, and during most of the downstroke the angle of attack is near 90°. At this high angle of attack, there is no distinct "leading edge" and "trailing edge" as far as the air flow is concerned, so one would expect the flow to behave similarly on both edges, with some differences resulting from the shape of the wing and feathering rotation.

Figure 3.20 shows the paths of particles released from the top surface of the wing during the second half of the downstroke (t/T = 2.17). These particle tracks represent the paths particles would take if the wing and flow field were "frozen" at this instant in time. They are colored by static pressure. The particle tracks show that there are vortical structures along the leading edge, trailing edge, and the root at this instant in time, and they are characterized by low pressure at the

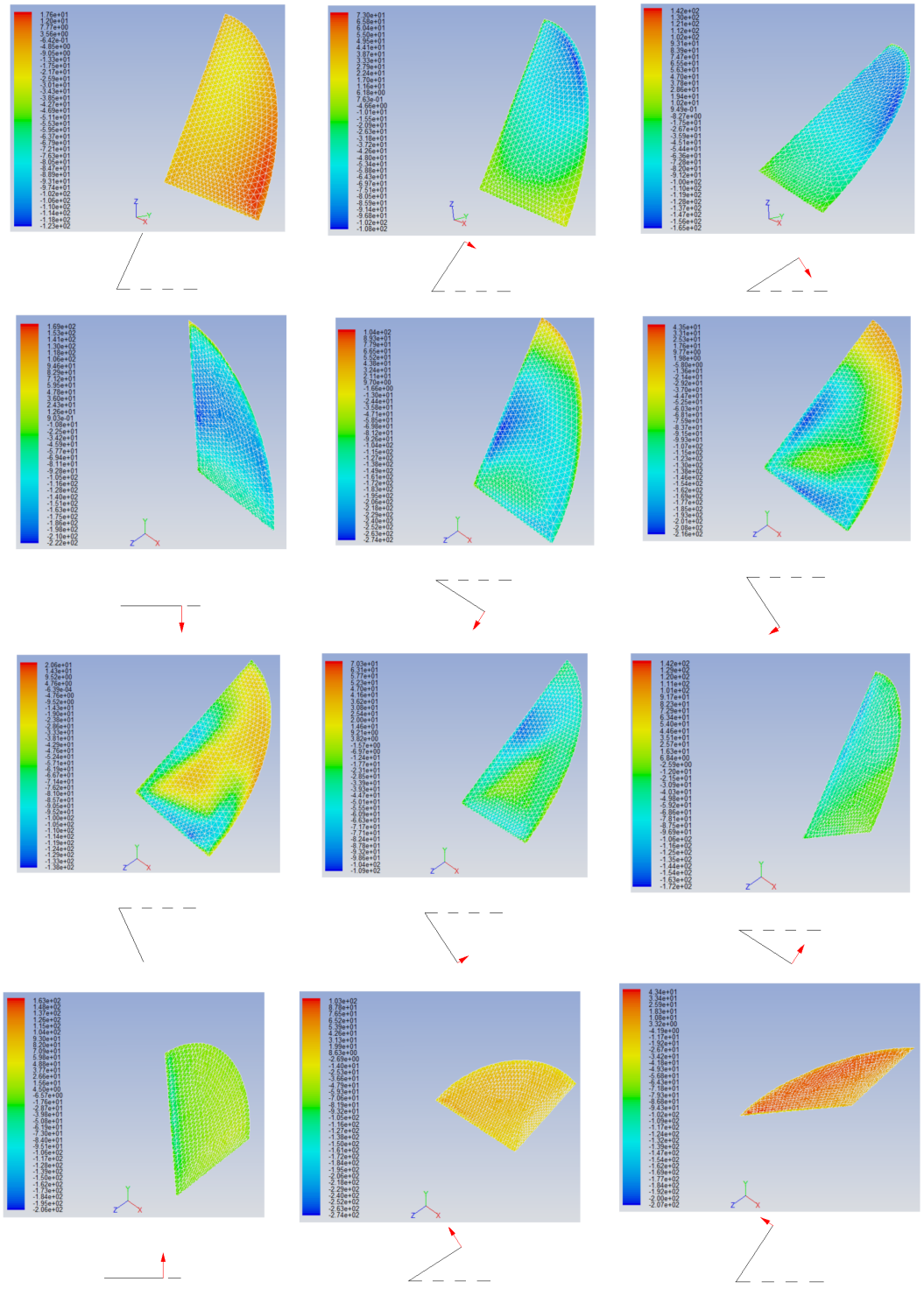


Figure 3.19: Pressure contours on the top surface of the wing

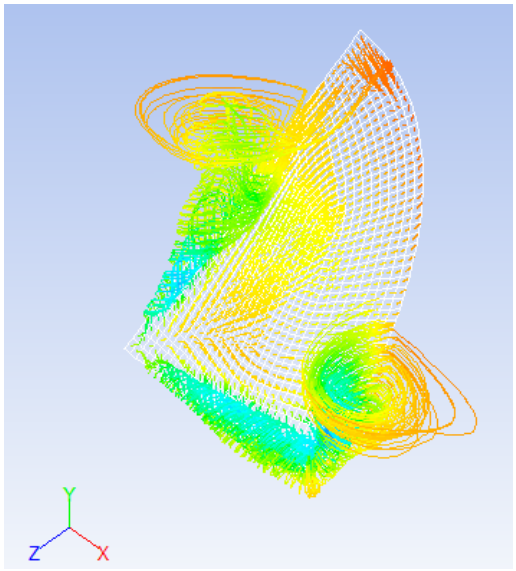


Figure 3.20: Vortical structures on wing's top surface

core. Both the leading-edge and trailing-edge vortices grow quickly in the spanwise direction and become weaker, with the effect being more severe on the trailing-edge vortex. Comparing this illustration with the corresponding surface-pressure plot in Fig. 3.19 (the middle image in the second row), it seems that these vortices— and especially the leading-edge vortex— may play a large role augmenting the low pressure on the top surface.

To further investigate the effects of these vortices, velocity vectors on planes normal to the wing at four spanwise sections are plotted in Fig. 3.21. These vectors are plotted relative to the flapping velocity of each section, because sections further out on the wing in the spanwise direction see a greater relative velocity than those nearer the root. This spanwise kinematic velocity gradient produces axial flow in the core of leading-edge vortices in insects, which is thought to be a major cause of their stability [6]. The location of each section is illustrated below the vector plot, and the vectors are colored by static pressure. The vectors are viewed from the root of the wing, so that the leading edge is on the left side of each image and the trailing edge is on the right side.

Figure 3.20 shows that the vortices on the top of the wing grow in size in the spanwise direction and eventually separate, at which point their pressure-reducing effect on the wing is weakened. This observation is confirmed by Fig. 3.21. As the distance in the spanwise direction is increased, both vortices grow in size and begin to separate from the surface of the wing. The leading-edge vortex remains stronger and attached farther in the spanwise direction, which explains why the pressure near the leading edge in the corresponding image of Fig. 3.19 is lower than near

the trailing edge. Figure 3.21 also shows that the vortices act to entrain air that would otherwise flow past the wing due to the separation at the leading and trailing edges, "dragging" it down with the wing and increasing the vertical force produced. This is especially true closer to the root, where the leading-edge and trailing-edge vortices are stronger and the relative flow speed is lower.

Though the QV is unique among MAVs in its use of drag to produce vertical force during hovering, this concept is not unprecedented in nature. Wang studied dragonfly flight using CFD and observed that, during hovering, the dragonfly uses a nearly vertical stroke plane and produces vertical force primarily via drag, just like the QV [57]. Wang's visualizations also show both a leading-edge vortex and a trailing-edge vortex. Wang also found that drag-based hovering can be as efficient as the lift-based hovering that many other insects use.

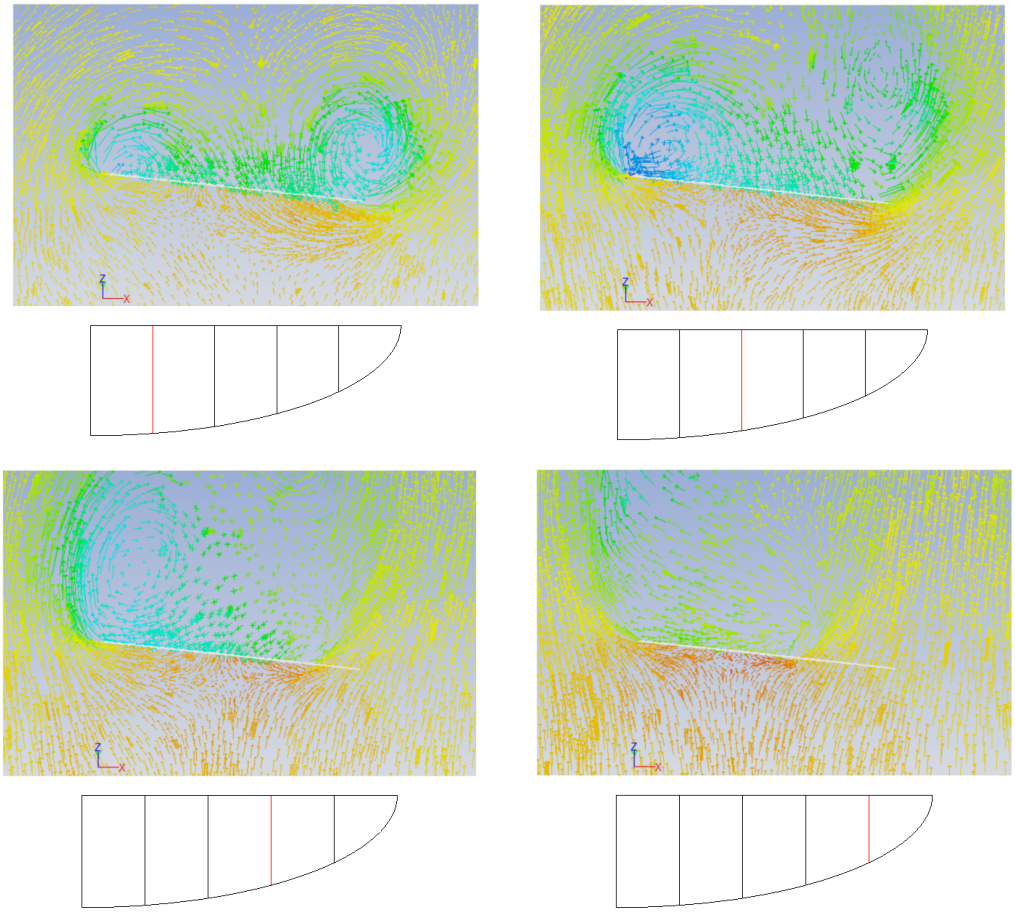


Figure 3.21: Velocity vectors colored by static pressure showing leading- and trailing-edge vortices during the downstroke

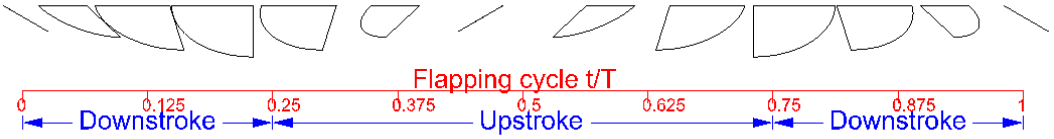
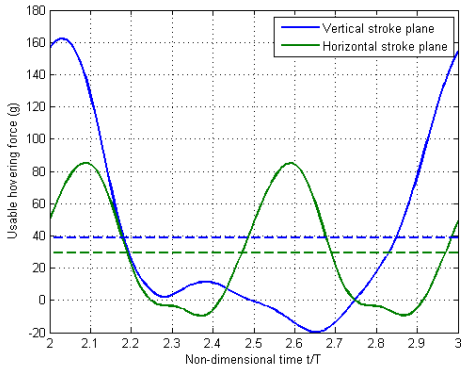
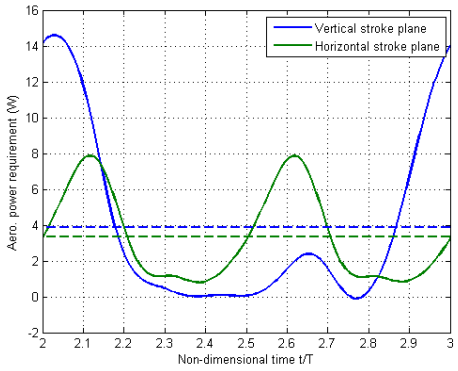


Figure 3.22: Wing motion illustration - horizontal stroke plane



(a) Usable vertical force (g)



(b) Aerodynamic power requirement (W)

Figure 3.23: Forces and power for vertical and horizontal stroke planes

Hovering with a horizontal stroke plane

In nature, many small flying creatures such as insects and hummingbirds are able to hover primarily via lift using a horizontal or nearly horizontal stroke plane. The exception seems to be the dragonfly, which hovers primarily via drag with a nearly vertical stroke plane, as was shown by Wang in his study [57]. The QV is designed with a vertical stroke plane, but because the horizontal stroke plane is so prevalent in nature, it is worth investigating that configuration alongside the vertical stroke plane.

A single test case with a horizontal stroke plane is performed here to compare against the QV design. Identical values of ϕ_{max} and f are used as in the "optimal" case with a vertical stroke plane ($\phi_{max} = 65.4°$, $f = 35$ Hz). However, with a horizontal stroke plane the feathering rotation must be symmetrical for the upstroke and downstroke to produce vertical forces via lift during each stroke. For this case, the total feathering angle θ_{max} is set to 120°, so that the angle of attack at the middle of each stroke is 30°. Equations 3.11 and 3.12 are still used as the rotational velocity equations. Figure 3.22 illustrates the wing kinematics for a flapping cycle, as seen from the side view.

The usable hovering force and aerodynamic power requirement are shown in Fig. 3.23 compared with the results for the vertical stroke plane case. As in previous simulations, the total forces and power requirement from the single-wing simulation

are multiplied by 4 for the Quad-Wing Vehicle. The dashed lines in the plots represent the cycle-averaged values. Note from Fig. 3.23(a) that the vertical force produced with a horizontal stroke plane (which, in this case, is lift) is symmetrical in the upstroke and downstroke. Some negative vertical force is produced during the rotational phases. The cycle-averaged vertical force for this case is 29.1 g, which is less than in the case with a near-vertical stroke plane, which produces 38.7 g. This is a significant difference, but it is not nearly as great as the difference in peak values, which are 85.2 g for the horizontal stroke plane and 162.3 g for the near-vertical stroke plane.

The aerodynamic power requirement trace for the horizontal stroke plane is also symmetrical in the upstroke and downstroke. The cycle-averaged power requirement for the horizontal stroke plane is 3.3 W, whereas for the near-vertical stroke plane it is 3.9 W. This is also a significant difference, but not greater than the relative difference in hovering force production for the two. What is of particular interest, however, is that the power requirement does not fluctuate nearly as much for the horizontal stroke plane. With a near-vertical stroke plane, the peak power requirement is nearly double that of the horizontal stroke plane case. A more constant power requirement should be easier on the actuators, both from the perspective of wear over time and the fact that lower peak output is needed.

One question that can be addressed when simulating hovering with a horizontal stroke plane is whether the leading-edge vortex acts on the wing in the CFD simulation as it has been shown in literature. With a vertical stroke plane, both a leading-edge and trailing-edge vortex are seen, but they work to increase the drag acting on the wing, not the lift. With a horizontal stroke plane, there should be only a leading-edge vortex which produces high lift by allowing the flow to reattach behind it on the wing. To investigate this, particles were tracked from the leading edge in the middle of the upstroke and are shown in Fig. 3.24. The tracks are colored by static pressure. This figure shows that there is a leading-edge vortex, with a low pressure core, which develops about 20% of the distance along the span and remains attached all the way to the tip. The flapping rotation direction is shown by the red arrow for reference.

Figure 3.25(a) shows the relative velocity vectors in a vertical plane perpendicular to the leading edge and situated 60% of the distance to the wingtip. The leading edge is on the right side of the image. The vectors are colored by vorticity magnitude in this case and clearly show the leading-edge vortex. Figure 3.25(b) is reprinted from Sane [21] as a comparison. The angle of attack of the wing in Fig. 3.25(a) is 30°; at this high angle of attack, the wing would be stalled in steady flow.

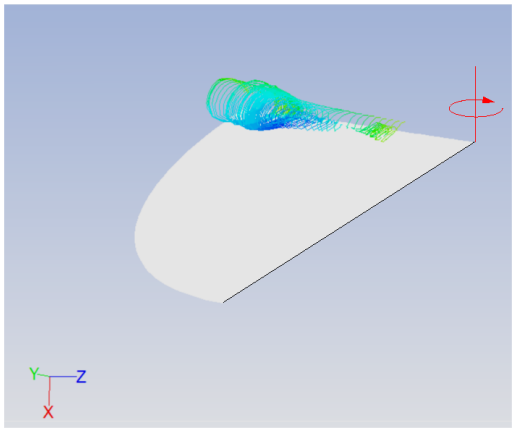
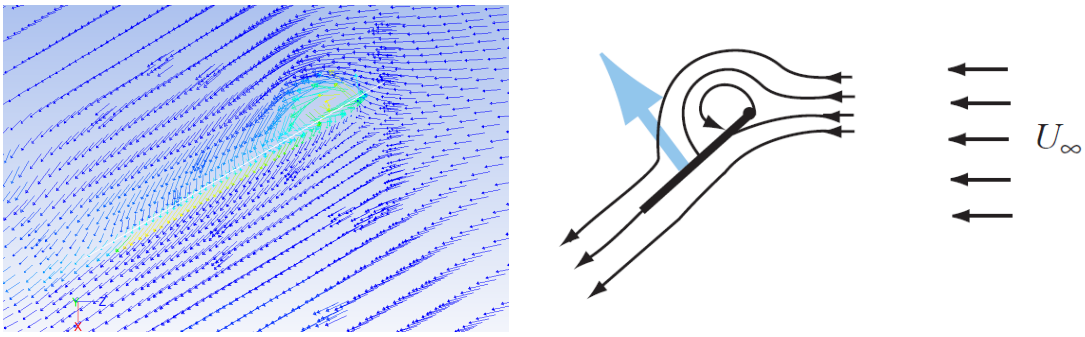


Figure 3.24: Leading-edge vortex for horizontal stroke plane



(a) Vectors showing the leading-edge vortex (b) Leading-edge vortex illustration

Figure 3.25: Leading-edge vortex: vectors in CFD and illustration from Sane [21]

The leading-edge vortex creates suction and allows the flow to reattach on the wing, producing a net circulation, downwash behind the wing, and thus lift. The vector field in Fig. 3.25(a) is very similar to the illustration by Sane in Fig. 3.25(b).

The case with the horizontal stroke plane does not outperform the case with the near-vertical stroke plane; the power requirement is lower, but the vertical force produced is also lower is also lower by a similar amount. However, it should be noted that the kinematics parameters for the horizontal stroke plane have not been optimized as they have been for the vertical stroke plane, and the feathering amplitude was chosen somewhat arbitrarily. Thus, it seems quite likely that better performance than is seen here could be realized via optimization of the kinematics parameters. The horizontal stroke plane does have some distinct benefits even if the performance on the basis of cycle-averaged vertical force and power is not better than for the near-vertical stroke plane. The first is that the fluctuations in force and power requirement are much smaller for the horizontal stroke plane, which is easier on the actuators and results in lower-magnitude body oscillations during the flapping cycle. Another benefit is that, since the feathering motion is symmetrical in both strokes, feathering would be easier to achieve passively via aeroelasticity.

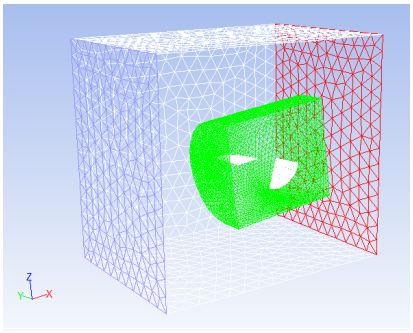
The concept of passive aeroelastic tailoring for flapping wings was explored by Frampton, et al. with good results [55].

3.4 Multi-Wing Modeling

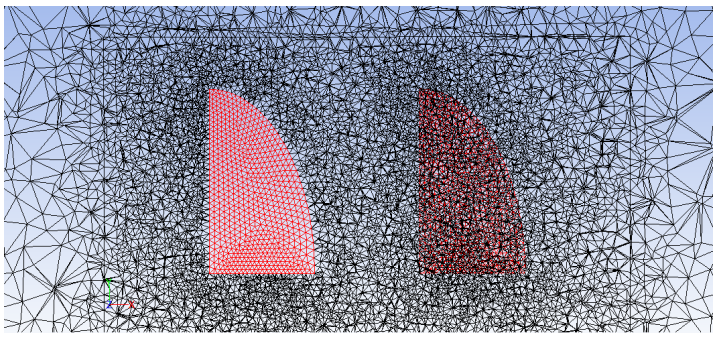
All of the CFD simulations performed in the previous sections were done with a single wing, and the results were multiplied by 4 to approximate the effect of all four wings in the Quad-Wing Vehicle. In the real Quad-Wing Vehicle, however, aerodynamic interactions between the wings may affect the forces produced. In this section, two wings are modeled—a forewing and a hindwing—to determine the affects of the aerodynamic interactions. It is still assumed that the wings on either side of the body have little effect on each other, since at their closest approach they are still separated by an angle of roughly 50° plus the width of the body in the optimal case. The forewings and hindwings, on the other hand, remain in close proximity throughout the flapping cycle, especially if they are flapping in phase.

The computational modeling of the two wings is similar to that used for the single-wing simulations. Both wings are contained in a single inner fluid zone, which in turn is contained inside an outer zone. The outer zone is bounded by a velocity inlet plane, an outlet plane held at atmospheric pressure, three side walls, and a symmetry plane. The computational domain is shown in Fig. 3.26(a). The preliminary design of the QV calls for the leading edge of the hindwing to be spaced about 1 chord length behind the trailing edge of the forewing. The two wings are close in proximity, so it is not possible for each wing to have its own structured fluid zone that rotates along with it, because the structured zones for the forewing and hindwing would collide with each other during the simulation. Instead, the inner zone bounds both of the wings and does not move. Both the inner and outer zone are discretized by unstructured tetrahedral elements, but only the inner zone, which has a finer mesh, deforms when the wings move. Figure 3.26(b) shows a section of the mesh in the plane of the forewing. The hindwing is visible as well, but it is not in the same plane as the forewing because the stroke plane inclination angle δ is applied to each wing.

The dimensions of the outer fluid zone are 46 cm in the x-direction by 32 cm in the y-direction by 42 cm in the z-direction. The inner fluid zone is a half cylinder joined with a rectangular box at its halving plane. The half-cylinder has a radius of 9 cm and is 20 cm in the x-direction, and the roots of the wings are against its halving plane. The rectangular portion joins the half-cylinder to the symmetry plane and is 4 cm in the y-direction. Thus, the inner zone is 13 cm total in that y-direction at its



(a) 2-wing CFD domain



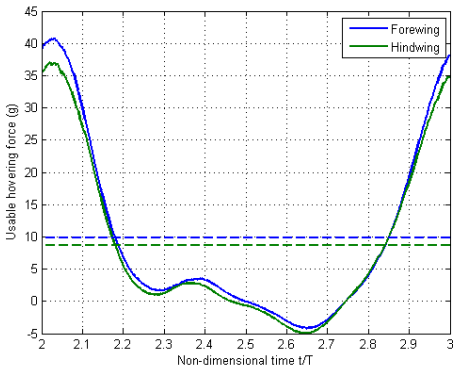
(b) Mesh in the forewing plane

Figure 3.26: Computational model of the forewing and hindwing

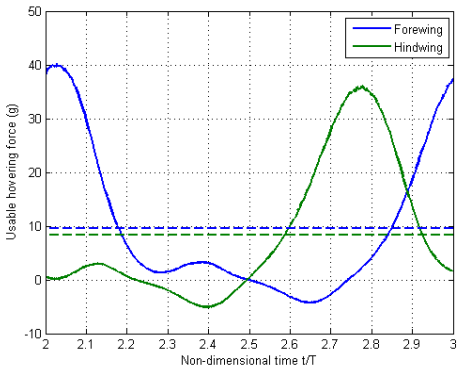
widest point. The entire domain has roughly 440,000 tetrahedral cells prior to mesh deformation. Since in these 2-wing simulations the fluid zone surrounding the wing must deform, Dynamic Meshing constraints are more stringent. The minimum size for local remeshing is initially set to the size of the smallest cell on the wing, while the maximum size is set to about 5 times that. The maximum cell skewness and spring constant factor are set the same as in the single-wing case. These settings actually increase the number of cells in the domain during the simulation, so that at the end of the simulation there are roughly 550,000 cells. No grid independence study is performed for the 2-wing simulations because the results show that the forces and power requirement on the forewing are very close to the results from the single-wing case in all the 2-wing simulations run.

Solver settings for the 2-wing simulations are identical to those used in the single-wing simulations, except for the time step size. Since Dynamic Meshing is required in the zone immediately surrounding the wing, the cells that must be deformed and remeshed are much smaller than in the single-wing simulations. If the same time step size were used as in the single-wing simulations, during some time steps the wing tip would move far enough to result in negative cell volumes, so for the 2-wing simulations the number of time steps per cycle is doubled. The additional time steps required, along with the greater number of cells in the 2-wing

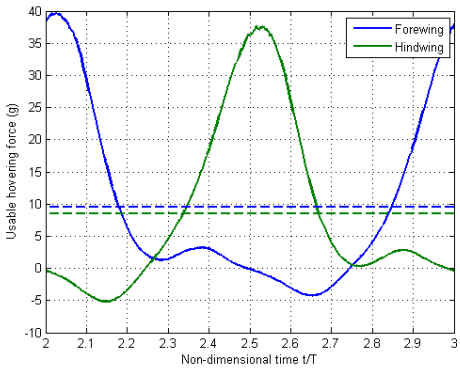
simulations, increases the computing time by 2.5 times, so each 2-wing simulation requires about 2 full days.



(a) 0° phase



(b) 90° phase

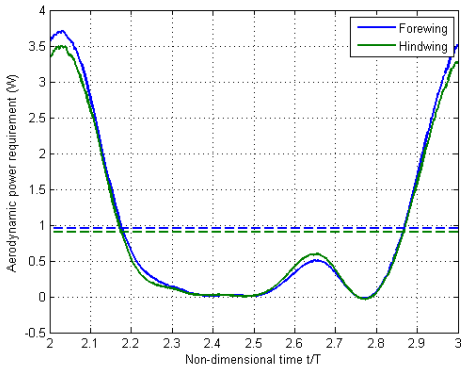


(c) 180° phase

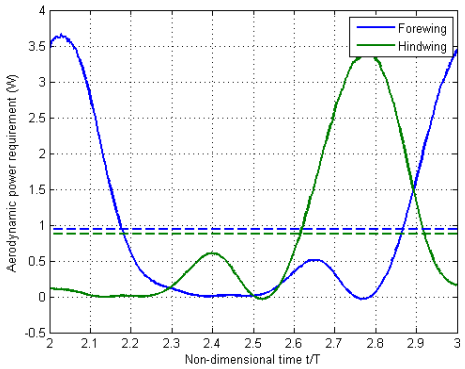
Figure 3.27: Usable hovering force for 2-wing simulations

In each 2-wing simulation,the "optimal" kinematics parameters from the single-wing sweeps are used: ϕ_{max} is 65.4°, f is 35 Hz, and θ_{max} is 90°. Three different simulations are performed, each using different relative phasing between the forewing and the hindwing. In each simulation, FLUENT reports the forces and moments throughout the cycle for each wing separately. Maybury and Lehmann performed a similar study of wing interactions in the dragonfly using experimental methods, investigating various phasings between the forewing and hindwing [53]. They found that in most cases, the hindwing's ability to produce vertical force was greatly reduced by the influence of the forewing, but when the hindwing led the forewing's motion by a phase of 90° its performance was mostly restored. However, the actual dragonfly uses different wing spacing and kinematics than the QV, and the wings' stroke plane is inclined more than the QV's, so those results may not be applicable to the QV. In the 2-wing study presented here, three distinct cases are presented: the wings in phase (0° phase difference), the hindwing leading the forewing by 1/4 cycle (90° phase difference), and the hindwing leading the forewing

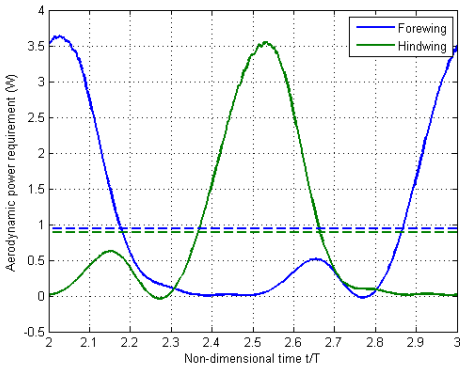
by 1/2 cycle (180° phase difference). The results of the simulations are shown in Figs. 3.27 and 3.28 for usable hovering force and aerodynamic power requirement, respectively. Each plot shows the traces and cycle-averages for the third full flapping cycle. The cycle-averaged forces and power requirements are shown by dashed lines.



(a) 0° phase



(b) 90° phase



(c) 180° phase

Figure 3.28: Aerodynamic power requirement for 2-wing simulations

Table 3.1: Cycle-averaged results of 2-wing simulations

Phase	F_h fore (g)	F_h hind (g)	P fore (W)	P hind (W)
0°	9.8	8.7	0.95	0.91
90°	9.5	8.3	0.94	0.88
180°	9.5	8.5	0.94	0.89
single wing	9.7	- - -	0.97	- - -

Table 3.1 gives the cycle-averaged hovering force and aerodynamic power requirement for each case and compares them to the single-wing case. As can be seen from this table and the figures, wing interactions do have some effect on the forces and power requirement, and this effect is most pronounced on the hindwing. In each case, the hindwing produces less hovering force and uses less power than the

forewing. The hindwing seems to produce the least force and use the least power in the case where it leads by 90° phase, but the relative differences between the cases are fairly small and may be within the range of computational error. Each case gives slightly different results for the forewing than in the single wing case, but the differences are, again, not of significant magnitude. The differences in performance between the forewings and hindwings, are significant, however.

For the design of the QV, it should be expected that the total hovering force production and power requirement will be slightly less than are predicted by the single-wing results, but the phase difference between the forewings and hindwings does not seem to be significant. This last result is convenient, because, at least from an aerodynamic perspective, there is no reason why the wings should need to be held in a constant phase relative to each other. The results shown here contrast with the study published by Maybury and Lehmann for a real dragonfly [53], but that should be expected since the configuration and kinematics are different. It should be noted, however, that the interaction effects in forward flight of the QV may be more significant than in hovering flight.

3.5 Discussion

This chapter presents aerodynamic analysis and optimization of the kinematics of the QV's flapping wings in hover using ANSYS FLUENT 12.1. Initially, 56 total simulations were performed sweeping the flapping amplitude and flapping frequency, and 24 additional simulations were then performed for a few representative cases sweeping the feathering amplitude. The results of these parameter sweeps show that, when hovering with a near-vertical stroke plane, it is the most conservative of aerodynamic power to use as high a flapping frequency as possible and setting flapping amplitude as required to achieve hovering flight. If the opposite approach is used—setting flapping amplitude to the max and then setting frequency to achieve hover—more power is required to overcome the aerodynamic moments. Physically, this result makes sense, because at very high flapping angles the wing produces mainly side force in the downstroke when hovering is achieved by means of a near-vertical stroke plane. For the QV design, if the actual wings are the same size as the wings used in the simulations, the actuators will need to allow for a resonant flapping frequency of at least about 27 Hz, but less power is be required if they can be driven at a higher frequency.

The effect of increasing feathering amplitude is to increase hovering force while simultaneously decreasing the aerodynamic power requirement. The hovering force

and power requirement reach optima when the feathering amplitude is about 90°. For the best case meeting the lift constraint of 35 g, increasing the feathering amplitude from 60° to 90° increased the hovering force from 35 g to 39 g and decreased the aerodynamic power requirement from 5 W to 3.9 W.

Two cases were performed to determine the effect of wing scaling on the forces produced and the aerodynamic power requirement during hover. In the first case, the wing was scaled so as to double the wing area, and in the second case the area was halved. In both cases, the forces produced were shown to be proportional to S^2 and the power requirement proportional to $S^{5/2}$, where S is the wing area. These relationships are consistent with the effect of scaling on the dynamic pressure seen by the wing during hover, and they will be important especially if the QV is to be scaled down in future prototypes.

Many insects and hummingbirds hover with the wings flapping in a nearly horizontal stroke plane. The QV is designed with a near-vertical stroke plane, but a case with a horizontal stroke plane was also simulated to compare against the QV design. In the simulated case, the total feathering angle was set somewhat arbitrarily to 120°, so that in the middle of each stroke the angle of attack is 30°. With a horizontal stroke plane, hovering force is produced equally in both strokes by means of lift, whereas with a horizontal stroke plane, the hovering force is produced almost entirely in the downstroke. The simulated case with a horizontal stroke plane produced 29.1 g of vertical force, which is about 25% less than the similar case with a vertical stroke plane, but it also required only 3.3 W of power, which is about 15% less than for the vertical stroke plane. The horizontal-stroke-plane simulation also exhibited a leading-edge vortex which augments the lift force by preventing stall at high angles of attack.

This comparison shows that hovering with a horizontal stroke plane does not necessarily give better performance than hovering with a vertical stroke plane; however, it should be noted that the kinematics for the horizontal stroke plane were not optimized at all, so it may likely be possible to achieve better performance with a horizontal stroke plane. The horizontal stroke plane does have some distinct advantages: the power requirement and forces have smaller fluctuations during the flapping cycle, which is easier on the actuators, and the symmetrical feathering motion can be achieved via passive aeroelastic tailoring of the wings as was done by Frampton, et al. [55]. Passive aeroelastic feathering would be harder to achieve with a vertical stroke plane because the feathering angle is not symmetric in the upstroke and downstroke. Because of these potential benefits, it would be worth investigating designs with a horizontal stroke plane further in future iterations of

the QV or for other MAVs.

Finally, simulations of both the forewing and hindwing were performed in CFD to investigate aerodynamic interactions between the two sets of wings. Three different phasings were used between the forewing and the hindwing: the wings in phase, the hindwing leading the forewing by 90°, and the hindwing leading the forewing by 180°. In each of the three cases, the forces and power requirement for the forewing were not changed much from the single-wing simulation, but the hindwing produced less hovering force and used less power to overcome the aerodynamic moments. At the wing spacing for the QV, the relative phase between the forewing and hindwing did not have a significant effect.

Chapter 4

Potential Flows Analysis

The previous chapter analyzes the aerodynamics of flapping wings for the Quad-Wing Vehicle in hovering mode using Computational Fluid Dynamics. CFD codes are very useful for the study of small-scale unsteady flight because they solve the discretized Navier-Stokes equations in order to resolve the flow field. However, due to their complexity, CFD solutions are also very computationally expensive and thus can be tedious for design work. In the previous chapter, for example, simulation of three full cycles of a single flapping wing required 19 hours on a relatively fast desktop computer, and a similar simulation of two flapping wings required 48 hours. Additionally, CFD solutions require extensive pre-processing in the form of grid generation and grid independence studies. For design work where hundreds of grids and solutions may be required, the process clearly requires a large amount of time. Because of this, a simplified solution approach would be greatly beneficial for design engineers.

One such approach is called the panel method, which is based on potential flow theory. In potential flows analysis, two simplifications—incompressible and irrotational flow—are made to result in a governing equation of flow that is linear and homogeneous. The linear nature allows the solution of the full flow field to be constructed by superposition of elementary solutions. Using panel methods, it is usually sufficient to construct the solution by modeling only the surface(s) of interest and the wake instead of the entire flow field, which greatly reduces the computational requirement. Panel methods are also attractive compared to other simplified methods, such as quasi-steady modeling coupled with blade-element theory, because they are able to model fully unsteady and 3D aerodynamic effects. However, the assumptions of incompressible and irrotational flow reduce the applicability of the method. The purpose of this chapter is to investigate the use of a 2D unsteady panel method to model hovering flapping-wing Micro Air Vehicle flight

and to compare the results with CFD solutions. The results shed light on whether panel methods may be a useful tool for designers of MAVs that fly in this manner.

4.1 Theory and Panel Code Development

The potential flow method developed here makes use of the velocity potential Φ. The velocity potential is related to flow velocity as follows.

$$\vec{V} = \nabla\Phi \tag{4.1}$$

This definition automatically satisfies the requirement for irrotational flow, $\nabla \times \vec{V} = 0$, as is shown in Eq. 4.2.

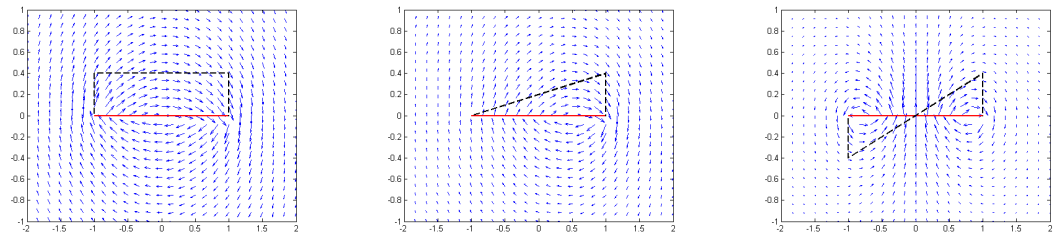
$$\nabla \times \vec{V} = \nabla \times (\nabla\Phi) = 0 \tag{4.2}$$

When the incompressibility requirement, $\nabla \cdot \vec{V}$, is applied to Eq. 4.1, the result is a linear, homogeneous partial differential equation which describes an irrotational and incompressible flow field, which is given in Eq. 4.3.

$$\nabla \cdot \vec{V} = \nabla^2\Phi = 0 \tag{4.3}$$

Since Eq. 4.3 is linear and homogenous, solutions can be constructed by superposition of elementary solutions. The panel method presented here follows methods described by Katz and Plotkin [58]. The flow field is constructed by superposition of elementary solutions on the airfoil surface and the wake. The airfoil surface is represented by a series of vortex panels with circulation strength that may vary linearly across each panel, while the wake is represented by discrete vortices shed from the trailing edge at the end of each time step. Owing to the type of surface formulation employed, the code is referred to as Unsteady Linear Vortex Panel Code, or ULVPC. Figure 4.1 illustrates the concept of vortex panels with linear-varying strength by showing velocity fields induced by vortex panels of varying strength distributions. The combined effect of all vortex panels on the airfoil surface creates a bound circulation distribution, $\gamma(s)$, on the airfoil, where s represents a location on the airfoil's surface. Each vortex in the wake has a circulation strength $\underline{\gamma}_l$, where l ranges from 1 to p and the subscript p represents the current time step. The numerical representation used in ULVPC is illustrated in Fig. 4.2.

Boundary conditions and initial conditions are required for the panel method. In potential flow problems, fluid "slips" along the surface of a solid body in the direction tangential to the surface. For a point on the surface moving with an



(a) Constant strength of 1 (b) Varying from 0 to 1 (c) Varying from -1 to 1

Figure 4.1: Illustration of vortex panels with linear-varying strength

arbitrary velocity $-\vec{v}$ in relative to an inertial reference frame, this surface boundary condition is represented mathematically by Eq. 4.4, in which $\hat{n}$ represents the surface normal direction.

$$\left(\vec{V} + \vec{v}\right) \cdot \hat{n} = 0 \tag{4.4}$$

The velocity of each point on the surface of the airfoil is dependent on the linear and rotational speed of the airfoil. Two separate coordinate systems are defined: the inertial system (X, Z) and the body system (x, z). The body axes move and rotate with the airfoil so that the airfoil surface location remains constant in the body coordinate system at all times. Any arbitrary airfoil motion in two dimensions may be realized by defining components of linear velocity $\vec{V}_a$ and rotational velocity Ω of the body origin, relative to the inertial coordinate system, as functions of time, after specifying an initial position and orientation. The time-marching procedure integrates the linear and rotational velocity functions to determine the position (X, Z) and orientation θ of the body system at any time, as shown in Fig. 4.3. By using these kinematics definitions and assuming that the airfoil is rigid and does not move relative to the body axes, the boundary condition of Eq. 4.4 can be written in the following form, as given by Katz and Plotkin. For the 2D panel code, $U\hat{i}$ and $W\hat{k}$ are the components of $\vec{V}_a$ and $\dot{\theta}\hat{j}$ is the only component of Ω.

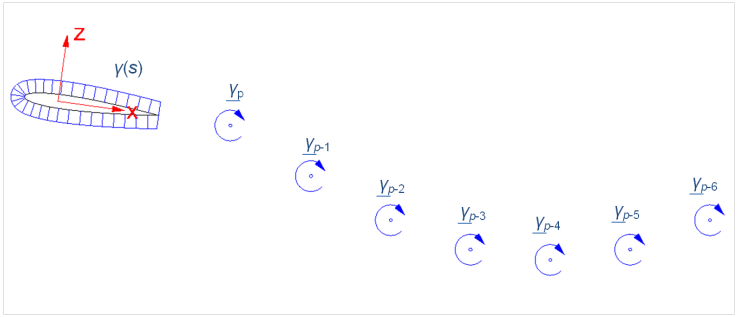


Figure 4.2: Numerical representation of the airfoil and its wake

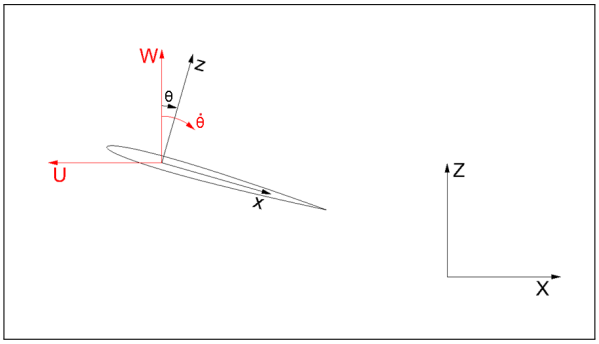


Figure 4.3: Definition of kinematics

$$\left(\nabla\Phi - \vec{V}_a - \vec{\Omega} \times \vec{r_a}\right) \cdot \hat{n} = 0 \tag{4.5}$$

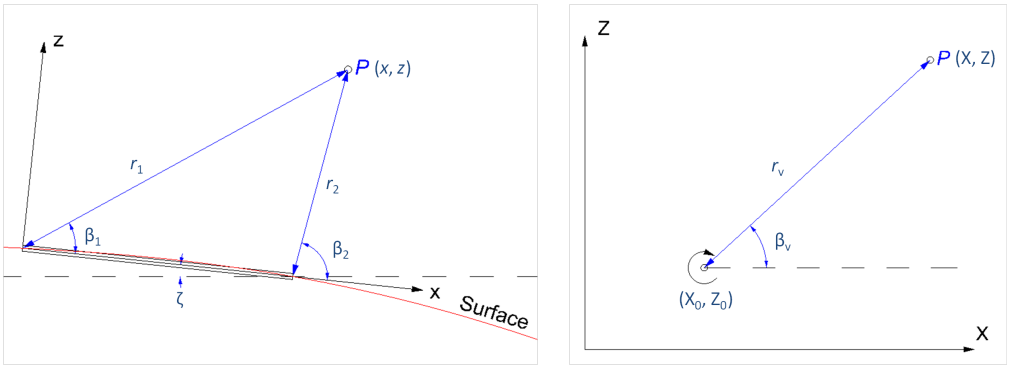
In Eq. 4.5, $\vec{r_a}$ is the vector from the origin of the body axes to a point on the airfoil surface. To satisfy this boundary condition on the airfoil surface, the airfoil is discretized into *collocation points* at which Eq. 4.5 is solved. There is one collocation point per vortex panel, usually placed at the midpoint of the panel. All terms in Eq. 4.5 are known from the user-defined kinematics and the geometry, except for $\nabla\Phi$ (which is equal to $\vec{V}$). In panel methods, $\nabla\Phi$ can be solved by superposition of the contributions from each vortex panel and wake vortex. To solve for $\nabla\Phi$ at each collocation point, a similar approach is taken here as is taken by Katz and Plotkin. First, consider the influence of a vortex panel of length d on an arbitrary point P located at (x, z). If the vortex panel is placed along the x-axis with its leftmost endpoint at the origin and has a strength $\gamma(x) = \gamma_0 + \gamma_1 x$, then the velocity potential, x-component of velocity, and z-component of velocity at point P are given by Eqs. 4.6, 4.7, and 4.8, respectively (these equations are given in Chapter 10 of Katz and Plotkin [58]).

$$\Phi_{pan} = -\frac{\gamma_0}{2\pi}\int_0^d \tan^{-1}\frac{z}{x-x_0}dx_0 - \frac{\gamma_1}{2\pi}\int_0^d x_0 \tan^{-1}\frac{z}{x-x_0}dx_0 \tag{4.6}$$

$$u_{pan} = \frac{\gamma_0}{2\pi}\int_0^d \frac{z}{(x-x_0)^2+z^2}dx_0 + \frac{\gamma_1}{2\pi}\int_0^d \frac{x_0 z}{(x-x_0)^2+z^2}dx_0 \tag{4.7}$$

$$w_{pan} = \frac{\gamma_0}{2\pi}\int_0^d \frac{x-x_0}{(x-x_0)^2+z^2}dx_0 - \frac{\gamma_1}{2\pi}\int_0^d \frac{x_0(x-x_0)}{(x-x_0)^2+z^2}dx_0 \tag{4.8}$$

Likewise, for a discrete vortex with strength $\underline{\gamma}$ placed at a location (X_0, Z_0) in the inertial frame, acting on an arbitrary point P, located at (X, Z) in the inertial frame, Eqs. 4.9, 4.10, and 4.11, give the influence on point P induced by the vortex.



(a) Vortex panel (b) Discrete vortex

Figure 4.4: Geometric definitions for solution of equations

$$\Phi_{disc} = -\frac{\gamma}{2\pi}\tan^{-1}\frac{Z-Z_0}{X-X_0} \tag{4.9}$$

$$u_{disc} = \frac{\gamma}{2\pi}\frac{Z-Z_0}{(X-X_0)^2+(Z-Z_0)^2} \tag{4.10}$$

$$w_{disc} = -\frac{\gamma}{2\pi}\frac{X-X_0}{(X-X_0)^2+(Z-Z_0)^2} \tag{4.11}$$

Details of the solutions of the integrals are given by Katz and Plotkin [58]. When using the solutions of these equations in a panel method, it is convenient to rewrite the panel strength distribution in terms of the endpoint panel strengths, γ_j and γ_{j+1}, instead of in terms of the left endpoint strength γ_0 and rate of change γ_1. Additionally, Katz and Plotkin employ several geometric definitions in the solution, including the relative angles between point P and the panel endpoints or the discrete vortex (β), the distances between point P and the panel endpoints or the discrete vortex (r), and the inclination of the panel relative to the inertial X-axis (ζ). These geometric definitions are illustrated in Fig. 4.4. The final equations derived from the previous six equations and the geometric definitions are given below.

$$\begin{aligned}\Phi_{pan} = &-\frac{\gamma_j}{2\pi}\Bigg(\left[x\beta_1-(x-d)\,\beta_2+z\ln\frac{r_1}{r_2}\right]\ldots\\ &-\frac{1}{d}\left[xz\ln\frac{r_1}{r_2}-\frac{zd}{2}+\frac{x^2-z^2}{2}\beta_1-\frac{x^2-z^2-d^2}{2}\beta_2\right]\Bigg)\ldots\\ &-\frac{\gamma_{j+1}}{2\pi}\left(xz\ln\frac{r_1}{r_2}-\frac{zd}{2}+\frac{x^2-z^2}{2}\beta_1-\frac{x^2-z^2-d^2}{2}\beta_2\right)\end{aligned} \tag{4.12}$$

$$u_{pan} = -\frac{\gamma_j}{2\pi d}\left[z\ln\frac{r_2}{r_1} + (x-d)(\beta_2-\beta_1)\right] + \frac{\gamma_{j+1}}{2\pi d}\left[z\ln\frac{r_2}{r_1} + x(\beta_2-\beta_1)\right] \quad (4.13)$$

$$w_{pan} = -\frac{\gamma_j}{2\pi d}\left[(x-d)\ln\frac{r_2}{r_1} - d + z(\beta_2-\beta_1)\right] + \frac{\gamma_{j+1}}{2\pi d}\left[x\ln\frac{r_2}{r_1} + d - z(\beta_2-\beta_1)\right] \quad (4.14)$$

$$\Phi_{disc} = \tan^{-1}\beta_v \quad (4.15)$$

$$u_{disc} = \frac{\underline{\gamma}}{2\pi}\frac{Z-Z_0}{r_v^2} \quad (4.16)$$

$$w_{disc} = -\frac{\underline{\gamma}}{2\pi}\frac{X-X_0}{r_v^2} \quad (4.17)$$

Each of the above equations is linear in terms of the unknown circulation strengths. For brevity in future equations, each of the coefficients of the circulation strengths given above can be represented by a single variable, as given by the following equations.

$$u_{pan} = a_{pan}\gamma_j + b_{pan}\gamma_{j+1} \quad (4.18)$$

$$w_{pan} = c_{pan}\gamma_j + d_{pan}\gamma_{j+1} \quad (4.19)$$

$$u_{disc} = e\underline{\gamma} \quad (4.20)$$

$$w_{disc} = f\underline{\gamma} \quad (4.21)$$

$$\Phi_{pan} = g\gamma_j + h\gamma_{j+1} \quad (4.22)$$

$$\Phi_{disc} = q\underline{\gamma} \quad (4.23)$$

There is one final transformation equation that must be applied, however, because u_{pan} and w_{pan} are written in the panel coordinate system in Eqs. 4.18 and 4.19. The transformation to the inertial system is given by Eq. 4.24.

$$\begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix} = \begin{bmatrix} \cos\zeta & 0 & \sin\zeta & 0 \\ 0 & \cos\zeta & 0 & \sin\zeta \\ -\sin\zeta & 0 & \cos\zeta & 0 \\ 0 & -\sin\zeta & 0 & \cos\zeta \end{bmatrix} \begin{Bmatrix} a_{pan} \\ b_{pan} \\ c_{pan} \\ d_{pan} \end{Bmatrix} \tag{4.24}$$

Using the definitions presented in the equations above, the boundary condition Eq. 4.5 can be written for any collocation point on the airfoil. Consider an airfoil with n collocation points undergoing an unsteady simulation where p time steps have passed. In this case, the body axis is moving with kinematic velocity components U in the X-direction and W in the Z-direction and is rotating about the body-axis origin at a rate $\dot{\theta}$. For the collocation point in question, the surface normal direction is represented by n_x in the X-direction and n_z in the Z-direction. Equation 4.5 can then be expanded as follows.

$$\begin{aligned} &\sum_{j=1}^{n} \left[\left(a_j \gamma_j + b_j \gamma_{j+1} \right) n_x + \left(c_j \gamma_j + d_j \gamma_{j+1} \right) n_z \right] \ldots \\ &+ \sum_{l=1}^{p} \left[e_l \underline{\gamma}_l n_x + f_l \underline{\gamma}_l n_z \right] - \left(U + \dot{\theta} r_z \right) n_x - \left(W - \dot{\theta} r_x \right) n_z = 0 \end{aligned} \tag{4.25}$$

The details of the wake play an important role in the solution of Eq. 4.3. Initially, at t=0, it is assumed that no wake exists. During subsequent time steps, a discrete vortex is shed into the wake along the path swept by the trailing edge. ULVPC places the shed vortex 30% of the distance between the current location of the trailing edge and the location from the previous time step. This is within the range suggested by Katz and Plotkin and gives good agreement with CFD and other panel codes (see Section 4.2). After the initial time step, the strength of each vortex shed into the wake is determined by Kelvin's theorem, which states that the total circulation Γ in the flow must be conserved.

$$\frac{\mathrm{D}\Gamma}{\mathrm{D}t} = 0 \tag{4.26}$$

The total circulation is comprised of the bound circulation on the airfoil surface, $\gamma(s)$, and the circulation contained in the wake, $\underline{\gamma}$, which for the panel code is in the form of discrete vortices shed at each time step. After p time steps have passed, the total circulation is defined by Eq. 4.27.

$$\Gamma_p = \left(\oint \gamma(s)\, ds \right)_p + \sum_{l=1}^{p-1} \underline{\gamma}_l + \underline{\gamma}_p \tag{4.27}$$

At the time step $p-1$ (the previous time step), the total circulation has the same form but lacks $\underline{\gamma}_p$ and the bound circulation distribution may be different; however, wake vortices are assumed to retain their original strength for all time. In order to satisfy Eq. 4.26, the total circulation must remain constant; thus $\Gamma_p - \Gamma_{p-1}$ is equal to zero. The result of this subtraction is given by Eq. 4.28.

$$\left(\oint \gamma(s)\, ds\right)_p - \left(\oint \gamma(s)\, ds\right)_{p-1} + \underline{\gamma}_p = 0 \tag{4.28}$$

When the bound circulation distribution is discretized by linear-varying vortex panels, Eq. 4.28 takes on the following form.

$$\left(\sum_{j=1}^{n+1} \gamma_j\right)_p - \left(\sum_{j=1}^{n+1} \gamma_j\right)_{p-1} + \underline{\gamma}_p = 0 \tag{4.29}$$

The airfoil is made up of n vortex panels, and there are 2 unknown panel endpoint strengths for each panel, so it would appear that application of Eq. 4.25 at each collocation point would give n equations with $2n+1$ unknown variables. However, the bound circulation distribution should be continuous on the airfoil surface; therefore, consecutive panels should share a single circulation strength at the connected endpoints. This reduces the number of unknowns to $n+2$ instead of $2n+1$. Equation 4.29 gives another equation, so there are now $n+1$ equations and $n+2$ unknowns. The final equation required to solve for all circulation strengths is the Kutta condition, which states that there is no discontinuity in the flow between the top and bottom of the trailing edge. This means the circulation at the trailing edge must be equal to zero. The airfoil input format of ULVPC requires that the first endpoint of panel 1 and the second endpoint of panel n come together at the trailing edge. Therefore, for ULVPC, The Kutta condition is represented by Eq. 4.30.

$$\gamma_1 + \gamma_{n+1} = 0 \tag{4.30}$$

Equations 4.25, 4.29, and 4.30 can be constructed in a matrix equation having the form $AB = C$, where A is an $n+2 \times n+2$ matrix consisting of the coefficients of all γ terms and $\underline{\gamma}_p$, B is an $n+2 \times 1$ vector containing all γ terms and $\underline{\gamma}_p$, and C is an $n+2 \times 1$ vector containing all known terms depending on the kinematics and the geometry. The forms of A, B, and C are given by the three equations below.

$$A = \begin{bmatrix} a_{11}n_{x1} + c_{11}n_{z1} & (a_2 + b_1)_1 n_{x1} + (c_2 + d_1)_1 n_{z1} & \dots & b_{n1}n_{x1} + d_{n1}n_{z1} & e_{k1}n_{x1} + f_{k1}n_{z1} \\ a_{12}n_{x2} + c_{12}n_{z2} & (a_2 + b_1)_2 n_{x2} + (c_2 + d_1)_2 n_{z2} & \dots & b_{n2}n_{x2} + d_{n2}n_{z2} & e_{k2}n_{x2} + f_{k2}n_{z2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{1n}n_{xn} + c_{1n}n_{zn} & (a_2 + b_1)_n n_{xn} + (c_2 + d_1)_n n_{zn} & \dots & b_{nn}n_{xn} + d_{nn}n_{zn} & e_{kn}n_{xn} + f_{kn}n_{zn} \\ 1 & 0 & \dots & 1 & 0 \\ d_1/2 & (d_1 + d_2)/2 & \dots & d_n/2 & 1 \end{bmatrix} \tag{4.31}$$

$$B = \begin{Bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_n \\ \gamma_{n+1} \\ \underline{\gamma}_p \end{Bmatrix} \tag{4.32}$$

$$C = \begin{Bmatrix} \left(U + \dot{\theta}r_{z1}\right) n_{x1} + \left(W - \dot{\theta}r_{x1}\right) n_{z1} - \sum_{l=1}^{p-1} \left(e_l n_x + f_l n_z\right)_1 \underline{\gamma}_l \\ \left(U + \dot{\theta}r_{z2}\right) n_{x2} + \left(W - \dot{\theta}r_{x2}\right) n_{z2} - \sum_{l=1}^{p-1} \left(e_l n_x + f_l n_z\right)_2 \underline{\gamma}_l \\ \vdots \\ \left(U + \dot{\theta}r_{zn}\right) n_{xn} + \left(W - \dot{\theta}r_{xn}\right) n_{zn} - \sum_{l=1}^{p-1} \left(e_l n_x + f_l n_z\right)_n \underline{\gamma}_l \\ 0 \\ \left(\sum_{j=1}^{n+1} \gamma_j\right)_{p-1} \end{Bmatrix} \tag{4.33}$$

After constructing these matrices, B can be solved by inverting A and left-multiplying it by C. Note that, for a rigid airfoil, the upper-left $n + 1 \times n + 1$ portion of the A matrix is independent of the position, orientation, and velocity of the airfoil, so significant computational time can be saved by computing these values only once. The reason they remain constant is because they are the coefficients of the airfoil's bound-circulation influence on the airfoil itself, and as long as the airfoil geometry remains constant, these coefficients also remain constant.

With the unknown circulation strengths in the B vector solved, the flowfield solution can be constructed. The velocity potential and velocity at any point in the field can be determined using Eqs. 4.12 through 4.24 and summing contributions from all vortex panels and wake vortices. ULVPC applies these equations at the collocation points to calculate the flow velocity and velocity potential on the airfoil surface. In potential flows, the Bernoulli formula is used to determine the pressure if the velocity is known. In unsteady flows, the Bernoulli formula contains an unsteady term related to the velocity potential and is given by Eq. 4.34.

$$\frac{p_\infty - p}{\rho} = \frac{1}{2}\left[\left(\frac{\partial \Phi}{\partial X}\right)^2 + \left(\frac{\partial \Phi}{\partial Z}\right)^2\right] + \frac{\partial \Phi}{\partial t} \tag{4.34}$$

In practice, a difficulty using Eq. 4.34 for a moving body arises; the pressure distribution on the surface of the airfoil is desired, but the surface of the body changes position and orientation from one time step to the next. To resolve this difficulty, Katz and Plotkin transform the Bernoulli equation to the body coordinate system, in which the position and orientation of the airfoil's surface remain constant, in Chapter 13 of their book [58]. The result of the transformation is given below.

$$\frac{p_\infty - p}{\rho} = \frac{1}{2}\left[\left(\frac{\partial \Phi}{\partial x}\right)^2 + \left(\frac{\partial \Phi}{\partial z}\right)^2\right] - \left(\vec{V}_a + \vec{\Omega} \times \vec{r_a}\right) \cdot \nabla \Phi + \frac{\partial \Phi}{\partial t} \tag{4.35}$$

The unsteady Bernoulli formula is significantly more complicated to use than the steady version because of the presence of the $\frac{\partial \Phi}{\partial t}$ term. To compute this term on the airfoil surface, ULVPC uses a first-order backwards difference stencil and uses Eqs. 4.12 and 4.15 to compute Φ.

Force components and pitching moment are determined by integration of the surface pressure. Usually, lift is defined perpendicular to the free-stream velocity and drag is defined parallel to it; however, for an arbitrary unsteady motion the free-stream direction may change constantly and thus it loses some of its meaning. In ULVPC, the net force is broken up into components in the vertical and horizontal directions relative to the inertial frame. For a hovering Micro Air Vehicle, the vertical force component is most important as this is the force that opposes gravity. The horizontal force is important for control and maneuvering. Horizontal force, vertical force, and moment about the quarter-chord (all per unit span) are defined by the following equations,

$$F_x' = -\oint p(s)\, \hat{n} \cdot \hat{i} ds \tag{4.36}$$

$$F_z' = -\oint p(s)\, \hat{n} \cdot \hat{k} ds \tag{4.37}$$

$$M_{qc}' = -\oint p(s)\, \vec{r}_{qc} \times \hat{n} ds \tag{4.38}$$

where, in Eq. 4.38, $\vec{r}_{qc}$ is a vector from the quarter-chord location of the airfoil to a point on the surface.

Following the computation of unknown circulation strengths, pressures, forces,

and moment, the flow field must be updated for the next time step. The airfoil's position, orientation, velocity, and angular velocity are all updated according to a user-defined kinematics function. The shape of the wake also must be updated. Since the wake has no mass and is not a solid surface, vortices shed into the wake must be free to move with the local flow field. The change in shape of the wake with time is known as *roll-up*. The roll-up is computed at the end of each time step by summing the velocity induced on each vortex in the wake by the bound circulation distribution of the airfoil and all other wake vortices. The change in position of each vortex in the wake for the next time step is equal to the product $\vec{V}_{induced}\Delta t$. Wake roll-up can have a significant effect on the solution. For certain airfoil motions, characteristics of thrust production and efficiency can even be inferred by the shape of the wake [47].

The algorithm used for ULVPC is shown in Fig. 4.5. ULVPC is programmed in MATLAB using a number of function files. The main function implements the algorithm shown in Fig. 4.5 and calls a number of exterior functions to perform the calculations. All of the required MATLAB functions, as well as an example set of commands to run ULVPC, are contained in Appendix B.

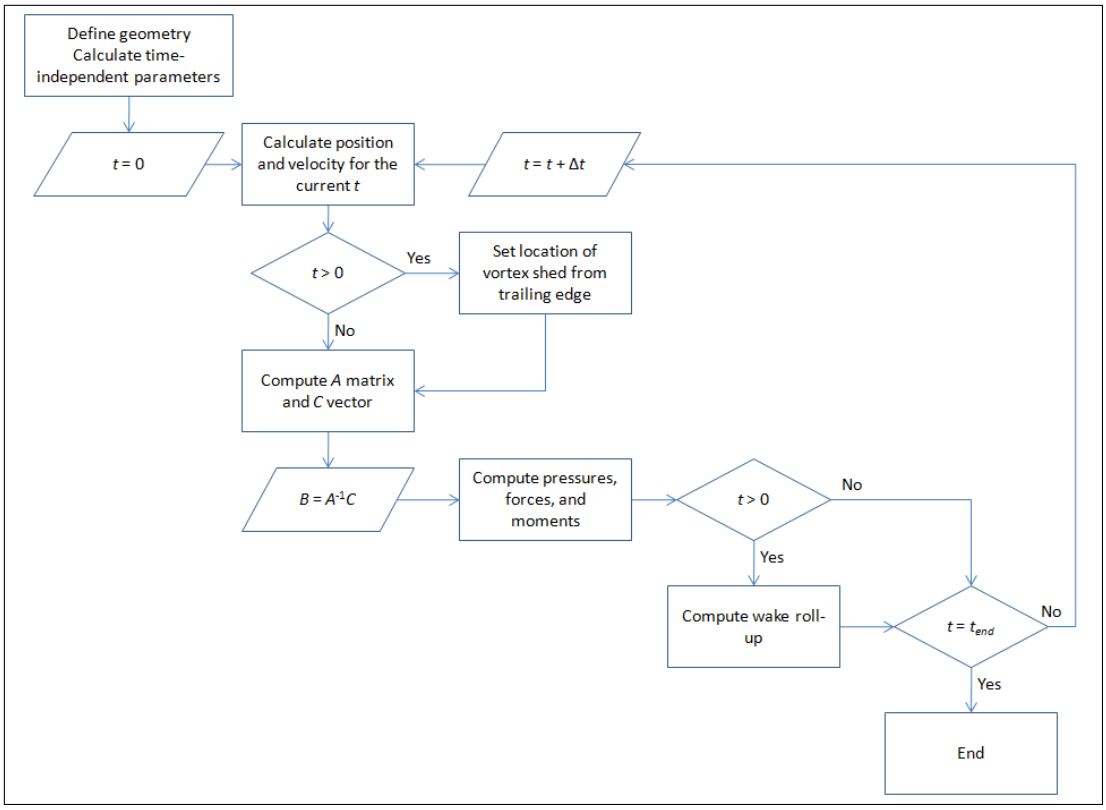


Figure 4.5: ULVPC algorithm flowchart

4.2 Panel Code and CFD Solver Validation

Jones, et al. [25] use a variety of numerical and experimental methods to investigate the use of flapping wings for propulsion of Micro Air Vehicles. These methods include 2D and 3D unsteady panel codes, 2D and 3D Navier Stokes and Euler solvers, and an experimental flapping setup for low (10^4) Reynolds number with variable aspect ratio wings. The wealth of data in this paper allows for validation of the panel code and 2D CFD solver used for the present work (data from the paper is used with permission). ANSYS FLUENT 12.1 is used as the CFD solver due to availability and its Dynamic Mesh capabilities which allow for unsteady motion of the airfoil (see Section 3.1.1). The solver includes viscosity and assumes the flow is laminar and incompressible throughout. For the validation study, a NACA 0014 airfoil oscillates sinusoidally in the vertical direction while moving to the left with constant velocity. Jones, et al. identify 2 major non-dimensional parameters that govern unsteady flows of this type: reduced frequency and non-dimensional amplitude. These parameters are defined as follows:

$$k = \frac{\omega c}{V_\infty} \tag{4.39}$$

$$h = \frac{A}{c} \tag{4.40}$$

where ω is the oscillation frequency, V_∞ is the upstream velocity or air, c is the chord, and A is the amplitude of the vertical oscillation. For the validation study, c is 1m, k is 0.2, and h is 0.4. V_∞ is chosen so that the Reynolds number is 10^6 for the viscous solver. The airfoil's motion (not drawn to scale) for one cycle is illustrated in Fig. 4.6.

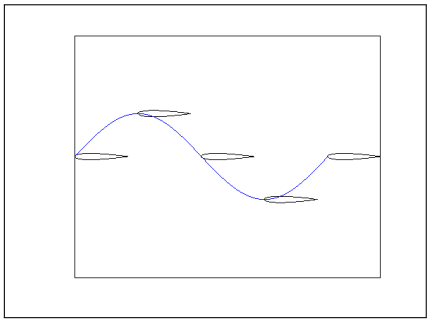


Figure 4.6: Illustration of kinematics for the validation study

The degree of unsteadiness for the validation case is rather low, and the fairly high Reynolds number combined with the low angle of attack throughout the trial ensures that the CFD result is free of large separation, preserving an apples-to-apples comparison between the numerical methods. This condition will likely not hold true for the later trials that compare the numerical methods at low Reynolds number appropriate for MAV flight, but nonetheless it can be used here to show that the panel code performs well in unsteady conditions when there is no separation present.

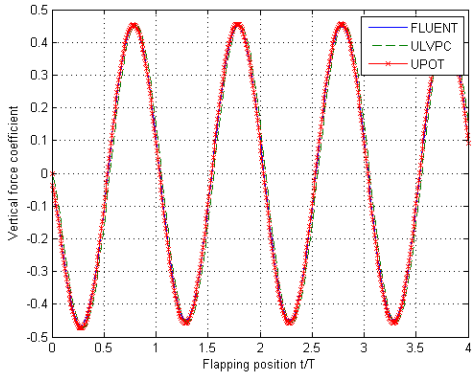


Figure 4.7: Comparison of vertical force

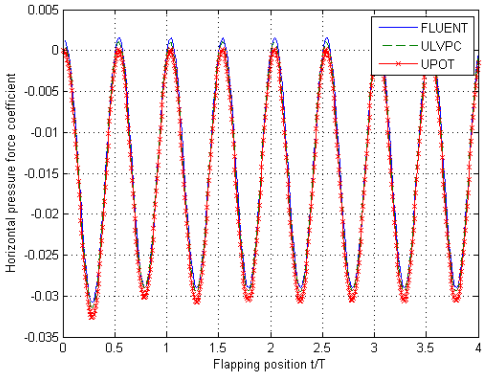
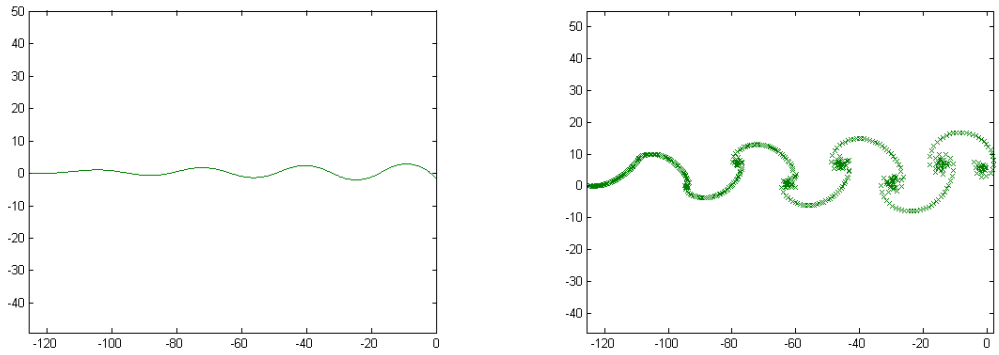


Figure 4.8: Comparison of horizontal force

Whether it will perform well enough at lower Reynolds number in flapping MAV flight is the main thrust of this chapter and the following comparisons.

For validation, the vertical and horizontal force coefficients returned by ULVPC and FLUENT are compared against the 2D panel code used by Jones, et al. (called UPOT; see Jones and Platzer [32]). Since FLUENT uses a laminar viscous solver, its force results contain contributions from viscous forces as well as pressure forces. In the validation case, viscous forces act as drag almost entirely in the horizontal direction. Due to the high Reynolds number the horizontal viscous forces remain small compared to the pressure forces, but nonetheless they are left out of the horizontal force trace for comparison purposes. Figures 4.7 and 4.8 show that all three solvers compare well in predicting both the vertical force and the horizontal pressure force, displayed as coefficients by normalizing with the upstream dynamic pressure. The vertical force coefficients are especially close for all three, though ULVPC exhibits a small phase shift compared to FLUENT and UPOT and both of the panel codes predict slightly high peak vertical force values compared to FLUENT. There is no phase shift apparent in the horizontal pressure force, but FLUENT is biased slightly in the positive direction compared to both the panel codes, though ULVPC is closer to the FLUENT result than is the UPOT result. Despite the slight differences, all three codes compare favorably and ULVPC and FLUENT may be used with confidence for the later unsteady trials.

Since ULVPC is coded in MATLAB, the built-in plotting functions make solution visualization relatively easy to achieve. One useful visualization technique is to look at the wake structure resulting from the unsteady motion. The wake is represented by discrete points whose position is updated and stored after each time step, so the wake's behavior can be easily observed by simply plotting the airfoil and wake after each time step. Figure 4.9(a) shows the wake structure trailing the airfoil after 4 complete cycles for the validation case. The airfoil is on the far left



(a) Validation study: $h = 0.4$ (b) $h = 4.0$

Figure 4.9: Illustration of wake roll-up

side of the figure but is too small to distinguish. Notice that the wake rolls up and grows in size as it moves downstream relative to the airfoil. This wake structure is indicative of thrust production, as momentum is convected downstream in the centerline of the wake [47]. Figure 4.9(b) shows the wake structure for a case with $h = 4.0$, which results in a higher level of unsteadiness and better illustrates the wake roll-up effect.

4.3 Non-Dimensional Parameters

The goal of this chapter is to determine under what conditions a panel code may be used as a design or analysis tool for hovering flight of a flapping Micro Air Vehicle. Since not all MAVs are of the same scale, the most general way of performing this study is in a non-dimensional fashion. Additionally, flow phenomena depend not on the absolute scale but on the values of these non-dimensional parameters. This concept of similarity allows, for example, wind tunnel experiments to be performed at a small absolute scale while ensuring the results will be applicable to the full-scale counterpart. Non-dimensional parameters are determined using Buckingham's Pi Theorem, a procedure by which dimensional parameters governing the flow are multiplied together with the appropriate powers so that the resulting group is non-dimensional [59]. The actual procedure is well-published, so it will not be repeated here, but the selection of the dimensional parameters and the resulting non-dimensional groups for hovering flight will be detailed.

4.3.1 Hovering Kinematics

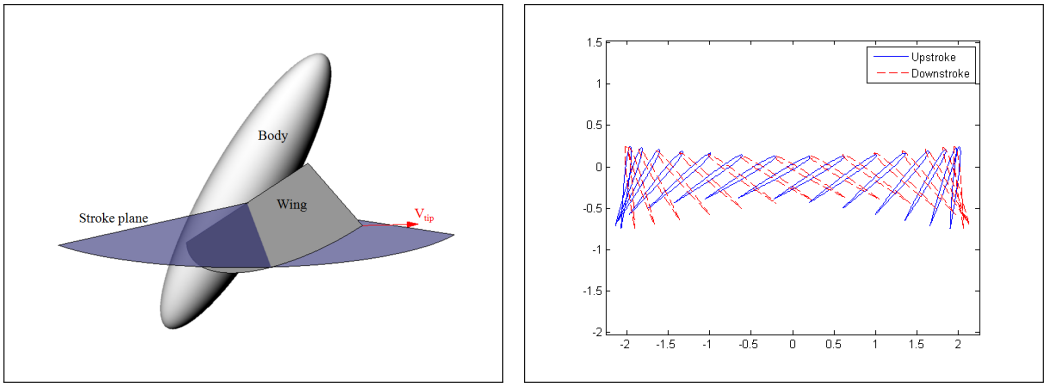
The forces produced in hovering flight are due in large part to the wing kinematics, especially in the 2D case where tip effects do not come into play. Sane and Dickinson [4], Ellington [60] and others note that most hovering creatures flap their wings in a horizontal or near-horizontal stroke plane, as shown in Fig. 4.10(a). The flapping cycle consists of two strokes—the upstroke and downstroke—during which the airfoil moves in a linear fashion. At the end of each stroke, the wing rotates in preparation for the ensuing stroke; these rotations are called pronation and supination. In general, the majority of the lift is produced during the translational phases while the primary purpose of the rotational phases, which are short in duration, is to change the angle of the wing in preparation for the ensuing stroke; however, Sane and Dickinson [5] suggest that the forces produced during rotation may also be important for maneuvering. Also, for an insect or hummingbird the inclination of the stroke plane may vary between the upstroke and downstroke, sometimes resulting in a figure-8 type of motion.

For the present study, the kinematics is simplified. The stroke plane is horizontal for all time, and the flapping velocity U and rotational velocity $\dot{\theta}$ are represented by simple sinusoids, with the center of pitching rotation placed at the quarter-chord location of the airfoil. For true hovering, the average velocity in the vertical direction would remain at 0, but for the purpose of generality a small vertical velocity W may also be considered. A is the distance traveled by the airfoil during half of a stroke. The angle of attack α is a function of the horizontal velocity, the vertical velocity, and the pitch angle. Because of this complication, the angle of attack during the middle of the translational stroke, called α_0, is selected as a parameter. Likewise, U and $\dot{\theta}$ vary during the cycle but are functions of the frequency ω and α_0, respectively. Figure 4.10(b) illustrates typical motion of the airfoil for a single cycle.

4.3.2 Π Groups

Of course, in addition to hovering kinematics, fluid properties are also important to force production. The most important are the density ρ and the viscosity μ. The chord, c, is the physical size of the airfoil and also impacts force production. We are left with a total of 7 parameters that govern the force production (per unit span) for a 2D wing in hovering flight.

$$F' = func\left(\omega, A, c, \alpha_0, W, \rho, \mu\right) \tag{4.41}$$



(a) Horizontal stroke plane (b) Illustration of kinematics

Figure 4.10: Motion illustrations

Including force as a parameter, there are 8 total dimensional parameters. There are 3 physical units: mass, length, and time. The Pi theorem dictates that there will be 3 repeating parameters used to construct 5 dimensionless Π groups. Using ω, c, and ρ as repeating parameters, the five groups are as follows.

$$\Pi_1 = \frac{F'}{0.5\rho\omega^2 c^4} \tag{4.42}$$

$$\Pi_2 = \frac{A}{c} = h \tag{4.43}$$

$$\Pi_3 = \alpha_0 \tag{4.44}$$

$$\Pi_4 = \frac{W}{\omega c} = \frac{1}{k} \tag{4.45}$$

$$\Pi_5 = \frac{\rho\omega c^2}{\mu} \tag{4.46}$$

The first and fifth Π groups are very similar in form to the traditional force coefficient and Reynolds number, respectively. In fact, if we replace ωc with $2\omega A/\pi = \bar{U}$ in both of these groups and apply some factors, we can rewrite these Π groups to represent the traditional force coefficient and Reynolds number, as shown below.

$$\Pi_1 = \frac{F'}{0.5\rho\bar{U}^2 c} \tag{4.47}$$

$$\Pi_5 = \frac{\rho\bar{U}c}{\mu} \tag{4.48}$$

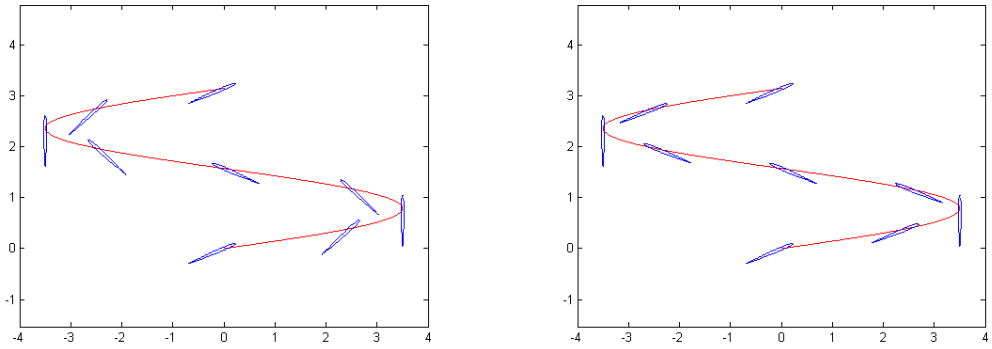
4.4 ULVPC vs. CFD for Hovering Flight

The validation study has shown that ULVPC performs well in cases with a small degree of unsteadiness and at high Reynolds number. At smaller sizes and higher degrees of unsteadiness, some of the assumptions made in developing the panel code begin to break down. Hovering flapping flight of a Micro Air Vehicle takes place in a regime that includes a high degree of unsteadiness and a low Reynolds number, where boundary layers tend to be large and there is a large amount of vorticity throughout the flow field. Thus, it may be expected that the panel code performance would be poor in this regime. However, for certain values of the non-dimensional parameters presented in the previous section, it is possible that the panel code may return results that are at least representative of the full Navier-Stokes solution, which would justify its use in the preliminary design stage of an MAV. In this section, two sets of comparisons representing two different types of flapping kinematics are performed for an airfoil undergoing a hovering motion. In each, the force coefficients generated by ULVPC are compared with those computed by FLUENT for a NACA 0006 airfoil with chord lengths of 1, 0.2, and 0.04 m. The smallest chord length is representative of the size of an MAV wing.

4.4.1 Kinematics Equations

The initial comparisons were performed using simple sinusoidal functions for the kinematics. The stroke plane is horizontal for all time. The angle of attack α_0 during the middle of the stroke is set to a fairly low value of 15°. The airfoil rotates about the quarter-chord location. In order to simulate hovering, Π_4 is set to 0.1. Note that for true hovering Π_4 is 0; however, with any real MAV there will be some small vertical velocities during a flapping cycle. A small positive value of Π_4 ensures the simulation still represents hovering flight while giving two benefits. The first is that it allows ω to be calculated from the Π_4 equation instead of being set arbitrarily, and the second is that it helps to convect the wake downwards, possibly avoiding numerical issues that may occur when the airfoil intercepts its wake made up of discrete point vortices. W is set arbitrarily to a value such that the airfoil travels one chord length vertically in 1 sec. The kinematics equations for the initial comparisons are as follows.

$$U = A\omega \cos(\omega t) \tag{4.49}$$



(a) Rotational motion 1 (b) Rotational motion 2

Figure 4.11: Rotational motions used for comparisons

$$\dot{\theta} = -\frac{B}{2}\omega \sin(\omega t) \tag{4.50}$$

In Eq. 4.50, B is the total angle of rotation to switch from one stroke to the next, which is determined by α_0. Using these equations of prescribed motion, though α_0 is set to a fairly low angle, there is nothing ensuring that the angle of attack remains small throughout the cycle. Large angles of attack at other points in the cycle may lead to boundary layer separation and hurt the quality of the panel code results. Given the same equations for the horizontal and vertical velocities, it is possible to set the rotation equation such that the angle of attack remains small throughout the cycle. The second set of comparisons between the panel code and CFD uses such a rotation prescription. The goal behind this comparison is to determine whether the agreement between the two numerical approaches improves when the angle of attack is held small at all times, which should reduce boundary layer separation. For this purpose, in the second set of comparisons, the initial angle of attack α_0 is set to 5°, but the instantaneous angle of attack varies sinusoidally from α_0 in the middle of each stroke to 0° at the end of each stroke. Equation 4.51 gives the resulting rotation equation.

$$\dot{\theta} = \alpha_0 \omega \sin(\omega t) - \frac{W A \omega^2 \sin(\omega t)}{[A\omega \sin(\omega t)]^2 + W^2} \tag{4.51}$$

Figure 4.11 illustrates the difference between the two rotation equations by showing snapshots of the airfoil and the direction of upstream velocity at different points in a cycle. In both figures α_0 is 15°. In Fig. 4.11(b) the angle of attack is always less than or equal to 15°, while in Fig. 4.11(a) it varies throughout the cycle and at some points is large. Note that these figures use Π_4 equal to 0.35 instead of 0.1 to make the illustrations more clear.

4.4.2 CFD Procedures

The same CFD techniques used to perform the validation study are repeated here. The airfoil is a thin symmetrical NACA 0006 airfoil, which is chosen for two reasons. The first reason is to represent a flapping MAV wing, which is typically thin and symmetrical. The second reason is that ULVPC is designed to handle airfoils having some thickness, and a better mesh in CFD can be realized by using an airfoil with some thickness as well. The NACA 0006 is thin enough that it can be representative of a flapping MAV wing but still has enough thickness to afford a good-quality mesh. A structured C-grid with fine resolution is created around the airfoil, and outside of that structured grid, the rest of the field is unstructured with triangular elements. The domain is wide enough that even in the most aggressive case the walls are always at least 10 chord lengths away from the airfoil, ensuring that the wall effect is not significant. The entire mesh is composed of roughly 20,000 triangular and quadrilateral cells before deformation. Figure 4.12 shows the mesh used.

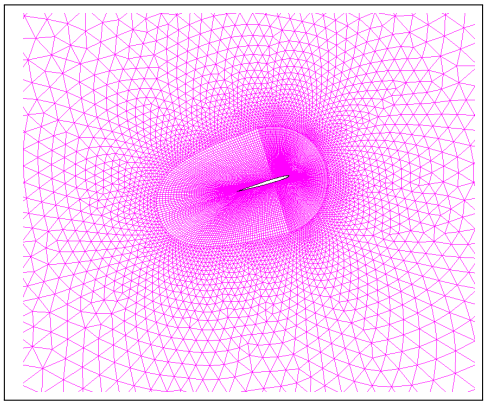
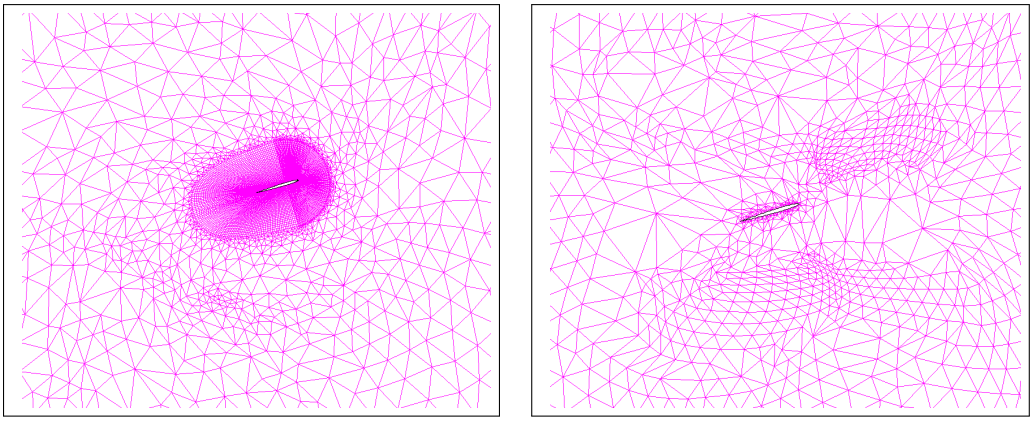


Figure 4.12: CFD grid used for comparisons

The CFD solver is unsteady and assumes viscous laminar flow. The solver is pressure-based and fully coupled in the pressure and momentum equations, which has given good results for the validation study and does not significantly affect the speed of calculations. For gradient evaluation, the Green-Gauss Node Based method is selected, which is recommended for unstructured grids in the FLUENT Theory Guide [56]. The QUICK scheme is used for discretization of the momentum equations, and standard pressure interpolation is used. This combination of algorithms has given good results and typically results in residues on the order of 10^{-6} or better for all equations during a time step. The time step depends on the flapping frequency and amplitude, which vary from case to case, but is set to give 120 time steps per cycle for the smallest values of h and is increased to up to 160 time steps per cycle for the largest value of h. The time advancement is first order implicit, which is the only option offered by FLUENT for the pressure-based solver.

FLUENT's Dynamic Meshing functions are used to allow for airfoil motion. As was done for the 3D flapping wing analysis in Chapter 3, spring-based smoothing and local remeshing are used to deform the unstructured portion of the mesh as the airfoil and inner structured zone move. For the comparisons presented in this



(a) Hybrid grid (b) Unstructured grid

Figure 4.13: Hybrid grid and unstructured grid after deformation

chapter, the spring constant for spring-based smoothing is set to a very low value of 0.001, the minimum size for remeshing is set to the length of the smallest cell in the unstructured zone, the maximum size is set to 7.5 times the minimum, and the maximum skewness for remeshing is set to the default value.

Figures 4.13 shows benefit of using a hybrid grid when Dynamic Meshing is required. Both images show the mesh after one full cycle at $h = 3.5$ and $\alpha_0 = 15°$. Figure 4.13(a) uses the hybrid grid approach, while Fig. 4.13(b) uses an unstructured grid for the whole domain. After one cycle, Fig. 4.13(a) shows that the hybrid approach allows the region near the airfoil, where gradients will be the highest, to maintain its high quality while the unstructured portion of the grid has lost some quality due to the Dynamic Mesh procedures. In Fig. 4.13(b), the grid quality has become severely degraded, especially near the airfoil where the greatest resolution is required. The degradation can be reduced somewhat by increasing the strictness of the local remeshing parameters, but this also greatly increases the number of cells in the grid, and the twisting of the grid caused by rotation will nonetheless hurt the quality no matter how strict local remeshing is set. In Fig. 4.13(a), the reduction of quality in the outer section is not as severe or critical because the cell size is larger and it is farther removed from the airfoil where gradients are highest.

4.4.3 Comparison of Force Coefficients and Flow Fields

For each of the two motion types, 12 total cases were compared. The goal is to compare the results first at larger size, where low Reynolds number effects should be minimal, and then decrease to finally compare the two codes at Micro Air Vehicle size. To accomplish this, the chord is reduced in each set of runs from 1 m to 4

Table 4.1: Values of non-dimensional parameters

Chord (m)	Π_2 (h)	Π_4 $1/k$	Π_5 (Re)
1	0.5, 1, 2, 3.5	0.1	2.2×10^5 - 1.5×10^6
0.2	0.5, 1, 2, 3.5	0.1	8700 - 61000
0.04	0.5, 1, 2, 3.5	0.1	350 - 2400

cm. The initial set of runs uses Eq. 4.50 for the rotational motion with α_0 equal to 15°. Table 4.1 shows the values of the important non-dimensional parameters used during these runs.

For each of the 12 cases, 4 complete cycles are run to avoid the effect of startup transients, and force data in the vertical and horizontal directions is compared for the fourth cycle. Each simulation nominally uses 120 time steps per cycle, but in CFD the number of time steps had to be increased at the highest value h because of Dynamic Mesh requirements, so that for $h = 3.5$ there were 160 time steps per cycle. The force traces from the panel code exhibited no significant differences when the time step was increased from 120 per cycle, even at the highest value of h. Forces are normalized to coefficients using Eq. 4.47. Figures 4.14 and 4.15 show the horizontal and vertical force coefficient traces for the fourth cycle, comparing ULVPC and FLUENT, with h increasing horizontally from 0.5 to 3.5 and c decreasing vertically from 1 m to 4 cm. These force traces are for the runs using motion type 1, which uses Eq. 4.50 for rotation. For each force trace presented, the beginning of the cycle occurs at the middle of the "upstroke" in which the airfoil moves from left to right. Stroke reversal occurs at t/T = 3.25 and 3.75. The blue force trace is the FLUENT result and the green force trace is the ULVPC result.

Both the shape and magnitude of the horizontal force traces given by ULVPC are fairly far off from the FLUENT result. There also seems to be a phase shift between the two, especially at lower values of h; at t/T = 3.5, at the middle of the "downstroke," the rotational velocity is zero so from a quasi-steady potential flow standpoint the horizontal force component should be zero as well. The presence of the wake from previous strokes is likely the cause of the phase shift, since at higher values of h, where the airfoil spends less time interacting with the wake from previous strokes, the phase shift diminishes.

Despite the poor agreement in shape and magnitude between the two codes, ULVPC in general does a good job capturing the average value of horizontal force, which is roughly zero. This result of zero net horizontal force is to be expected for hovering flight with a symmetrical motion and horizontal stroke plane. However, even the average of the individual strokes seems to be captured fairly well by the panel code. This is important because for an asymmetrical motion or non-horizontal

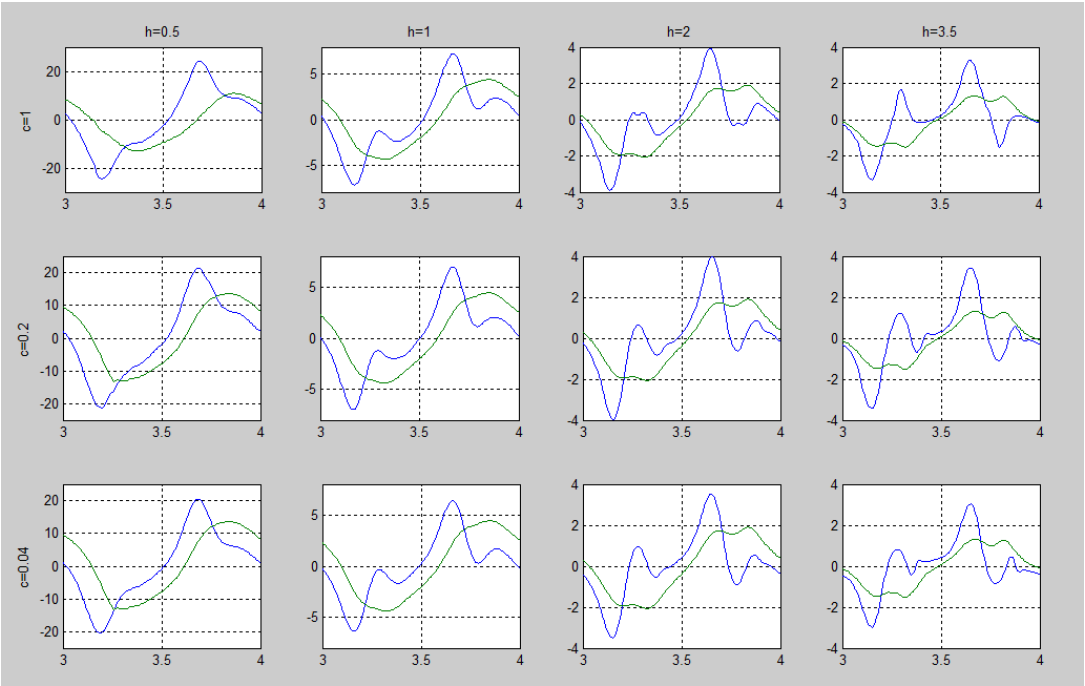


Figure 4.14: C_x versus flapping cycle t/T. Blue trace is FLUENT, green is ULVPC.

stroke plane, the overall average horizontal force may not be zero, and the stroke-averaged horizontal force plays a role in the MAV's body oscillations. Note that for the panel method, the force coefficients are independent of scale. For FLUENT, the coefficients are scale-dependent, but for this particular case there is only a minor change as the chord is decreased.

The comparison between the vertical force coefficient time traces for ULVPC and FLUENT is shown below. In general, the shape of the two is roughly the same but ULVPC under-predicts the vertical force coefficient. There is also a phase shift, as there was for the C_x traces, which decreases as h increases. For both codes, the maximum vertical force occurs prior to the middle of stroke reversal, when the airfoil is at a high angle of attack and is also rotating in a manner that causes a net upwash and increases the vertical force (this phenomenon is known as Kramer's effect [21]). The minimum vertical force occurs following stroke reversal when the airfoil is still at a high angle of attack but the rotational velocity creates a net downwash on the airfoil, decreasing the vertical force.

As h increases, the rotational speed increases at the stroke reversals, and the force traces from FLUENT show "bumpiness" beginning before the reversal points and continuing through reversal. ULVPC does not predict this behavior, suggesting that separation or vortex shedding is the likely cause for it. Figure 4.16 shows velocity vectors plotted by FLUENT for a range of t/T = 3.17 to 3.33 for $h = 3.5$ and $c = 1$. This captures the stroke reversal portion of the cycle, where translational

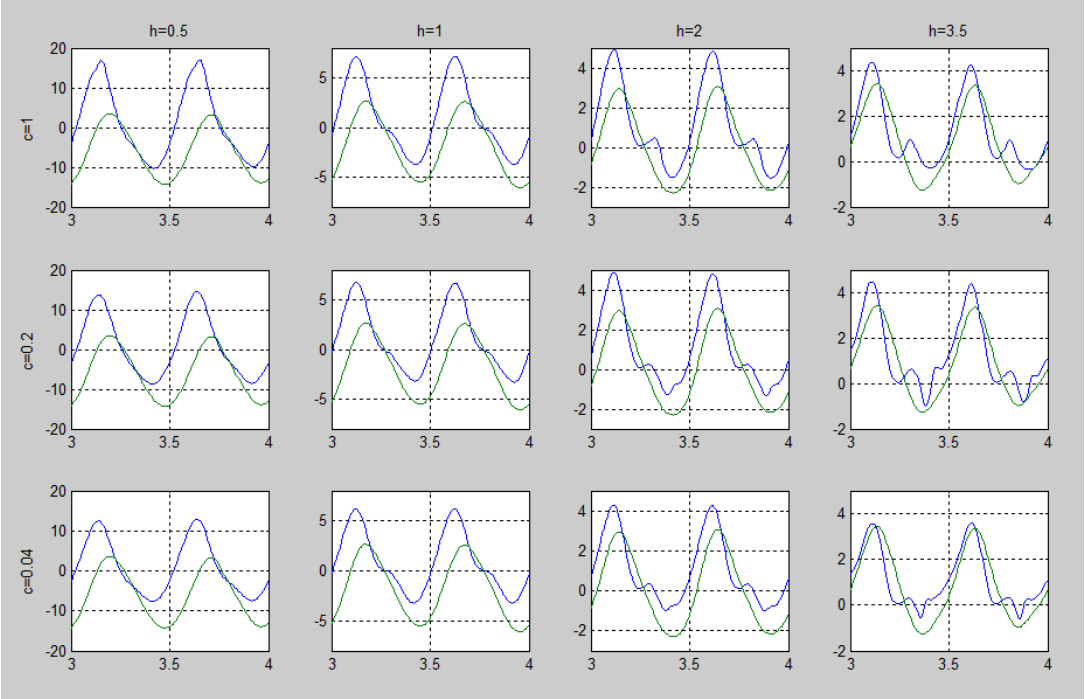


Figure 4.15: C_z versus flapping cycle t/T. Blue trace is FLUENT, green is ULVPC.

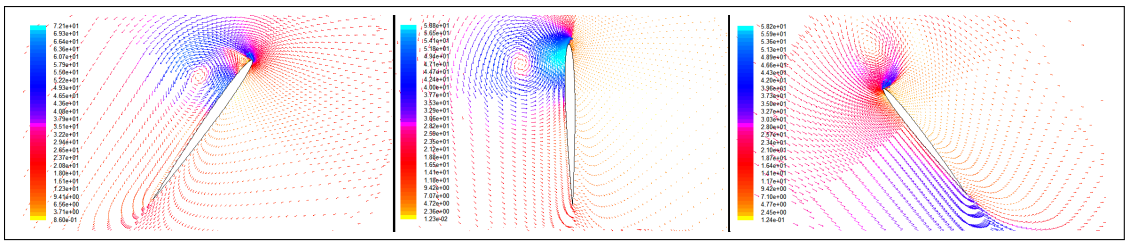
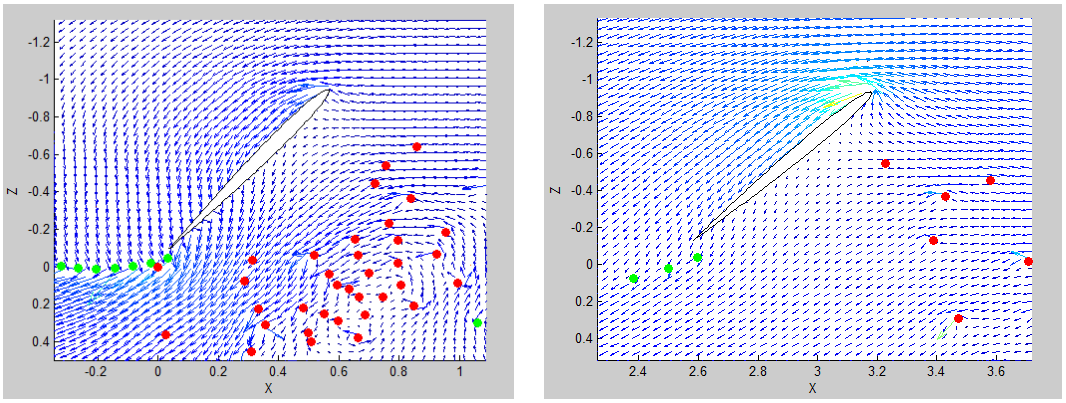


Figure 4.16: Vortex shedding during stroke reversal

flapping speed is small or zero and the rotational speed is high in the counterclockwise direction. The vectors are drawn relative to the airfoil's translational velocity. In FLUENT, vortex shedding, characterized rotating flow with a low-pressure core, occurs from the leading edge due to the high angle of attack. This vortex remains close to the leading edge of the airfoil throughout stroke reversal, and is the likely cause of the bumpiness in the force trace. Note also that at lower values of h there is no bumpiness in the FLUENT force trace, likely because the low flapping amplitude keeps translational and rotational speeds small enough that there is no vortex shedding.

Interestingly, the ULVPC result approaches the FLUENT result (except during stroke reversal) as h is increased and c is decreased. The effect of the wake has already been mentioned; since ULVPC matches FLUENT more closely at higher h where the previously shed wake is farther from the airfoil for most of the stroke, it seems logical that the effect of the wake in the panel code may be over-pronounced,



(a) ULVPC: $h = 0.5$, t/T=3.17 (b) ULVPC: $h = 3.5$, t/T=3.17

Figure 4.17: ULVPC vector fields before stroke reversal

at least when the airfoil passes close the wake shed in a previous cycle. Figure 4.17(a) shows velocity vectors computed by ULVPC at t/T = 3.17 for $h = 0.5$. This is the point in the cycle where ULVPC predicts the peak vertical force is reached. At $h = 0.5$, the previously shed wake has a strong effect on the forces produced throughout the cycle. The filled circles in the plot represent the discrete vortices shed into the wake from the trailing edge; red circles are vortices with counterclockwise rotation and green circles are vortices with clockwise rotation. Of particular interest are the counterclockwise vortices shed during the previous cycle, near the bottom surface of the airfoil. These vortices act to increase the velocity on the bottom surface, decreasing the pressure there and the vertical force produced. In fact, the clockwise vortices being shed into the wake during the current cycle augment this effect. In contrast, Fig. 4.17(b) shows the same picture for $h = 3.5$; in this case there are far fewer vortices from previous cycles in the nearby vicinity, bottom-surface velocity is not increased as much and the lift-reducing effect of wake interaction is not nearly as pronounced, as can be verified from the force traces. The unusually large size of some individual vectors near wake vortices is due to the singular nature of the discrete-vortex wake representation.

As chord decreases, the FLUENT vertical force traces are decreased in magnitude and actually very nearly match the ULVPC result for $h = 3.5$, $c = 4$ cm. The reason for this is not certain, but it is possible that vorticity shed during stroke reversal is more intense at lower Reynolds number and has a similar effect as the wake in the panel code. In general, though, the panel code can be expected to under-predict the vertical force, except for in cases with very high h, even at the size of a typical MAV wing.

The second set of comparisons uses Eq. 4.51 for rotational motion. The goal of this set is to determine whether better agreement between the two codes can be

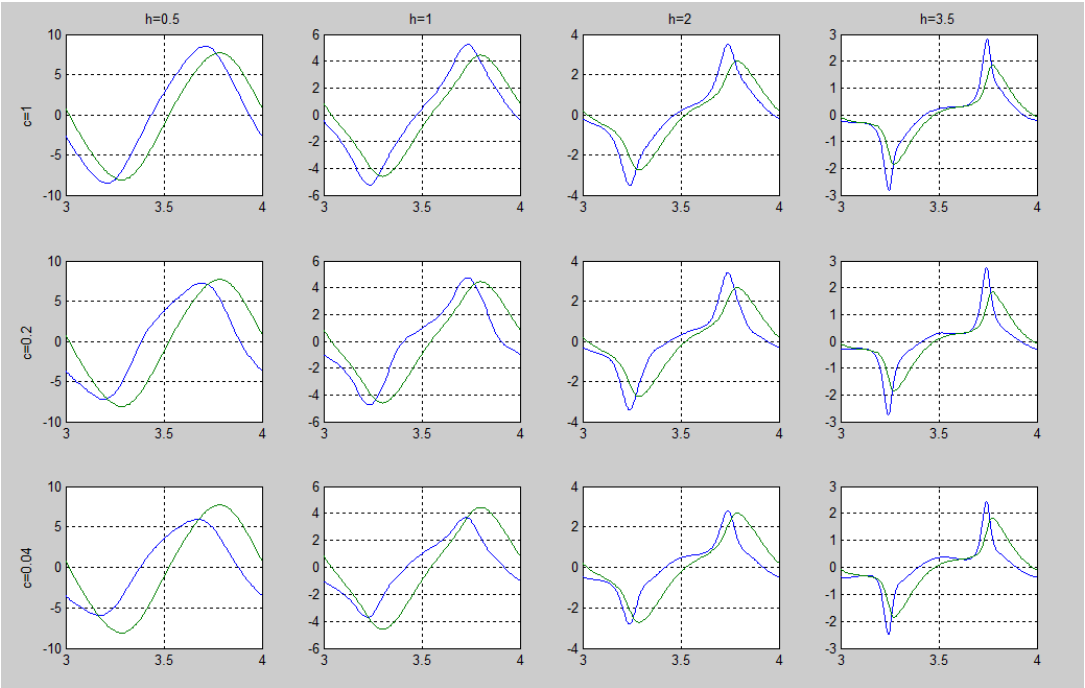


Figure 4.18: C_x versus flapping cycle t/T. Blue trace is FLUENT, green is ULVPC.

achieved when the angle of attack is kept small throughout the cycle. Equation 4.51 ensures that this is true, but to further discourage flow separation the initial angle of attack α_0 is set to only 5° in this case. This is also the maximum angle of attack for the second motion type. However, while this equation keeps the angle of attack small, it also results in very high rotational rates at the stroke reversals if Π_4 is kept small. This would likely cause separation. In order to avoid this scenario, Π_4 is increased to 1, which is still relatively small, but no longer describes hovering or near-hovering flight. Instead, it may represent a Micro Air Vehicle climbing at low speed. Nevertheless, the results will show whether separation due to high angle of attack in the first case is the major cause for the disparity in the results between the two codes. Other than the change in Π_4, the non-dimensional parameters for these runs are the same as shown in Table 4.1. Figure 4.18 shows the coefficients of horizontal force for these runs, with the FLUENT result in blue and the ULVPC result in green, as before.

The C_x results from ULVPC and FLUENT match up quite well for these runs. There is still a noticeable phase shift. For hovering flight, though, we are generally interested in the cycle-averaged forces and not the instantaneous forces, so the phase shift is rather inconsequential. In these cases, both the shape and magnitude of the force traces are fairly consistent between the two codes, though for high values of h FLUENT predicts a higher force spikes during stroke reversals. The magnitude of the spikes predicted by FLUENT decreases as c decreases, and ULVPC actually

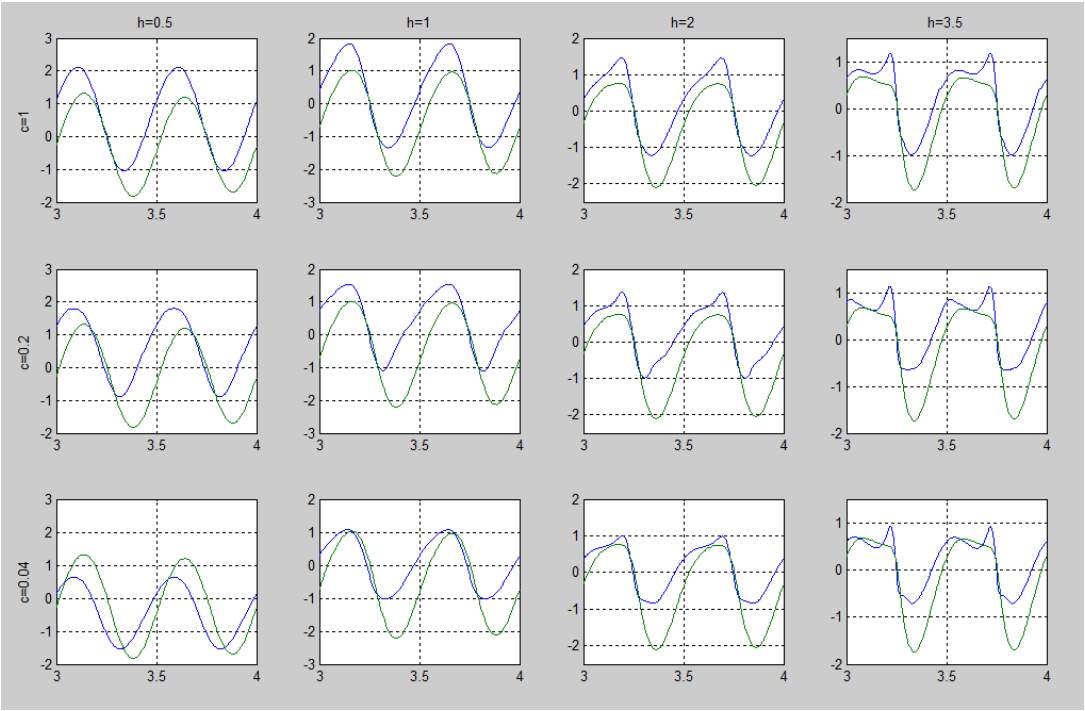


Figure 4.19: C_z versus flapping cycle t/T. Blue trace is FLUENT, green is ULVPC.

over-predicts the force peaks at the lowest values of h and c. At the exact middle of stroke reversal (t/T = 3.25 and 3.75), the FLUENT traces reach a peak in all cases. The phase shift in ULVPC causes the maximum horizontal force to lag behind stroke reversal to some extent, but in this case the lag does not seem to change much with changes in h or c. The phase lag in this case may have to do with the discrete representation of the wake; recall that in the validation study ULVPC exhibited a phase lag compared to FLUENT and UPOT. UPOT has a higher-order wake representation [32], so the phase lag is likely a product of the discrete wake and not a characteristic of panel codes in general. The phase lag does not impair ULVPC's ability to predict the average force production during flapping. In general, ULVPC does an admirable job matching full CFD result for horizontal force coefficient when the angle of attack is kept small, despite the fact that it experiences high rates of rotation during stroke reversals. Figure 4.19 shows the corresponding vertical force coefficients for these runs.

For this second type of motion where Π_4 is increased and the angle of attack is kept small, the agreement in vertical force coefficient between ULVPC and FLUENT is a little bit better than in the first type of motion, but not drastically better. There are some noticeable differences between the vertical force for this type of motion and the previous, however. First, there is no bumpiness in the FLUENT curve during stroke reversal, suggesting that there is no vortex shedding. This is confirmed by Fig. 4.20, which shows velocity vectors during the stroke reversal phase for $h = 2$,

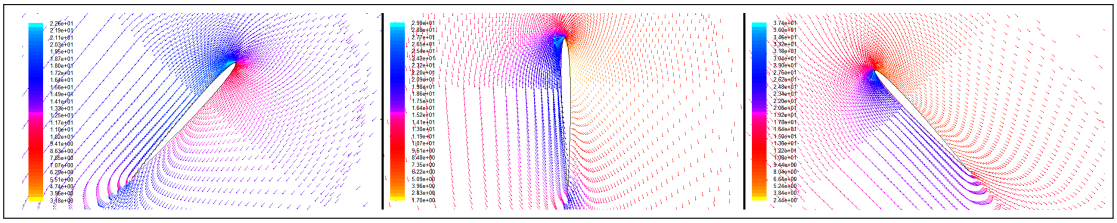


Figure 4.20: Stroke reversal free of vortex shedding

$c = 1$ from FLUENT. In contrast with Fig. 4.16, which depicted stroke reversal in the previous motion type, there is no vortex shedding present because the angle of attack is kept small. There is still a high degree of circulation produced by the airfoil's rotation.

Though there is no vortex shedding, the main differences between the FLUENT vertical force coefficient time history and the ULVPC time history still occur during the stroke reversal phase. Except for the smallest chord cases, FLUENT's vertical force at the beginning of stroke reversal is higher than ULVPC's, and ULVPC's negative force at the end of stroke reversal is greater in magnitude than FLUENT's. At the exact middle of stroke reversal (t/T = 3.25 and 3.75) when there is no translational velocity and rotational velocity is at its highest, both codes predict roughly zero vertical force.

However, ULVPC predicts a greater negative vertical force produced during and following stroke reversal, which deviate more from the FLUENT force trace as h is increased and c is decreased. This discrepancy may be due to boundary layer separation, which is not apparent from the velocity vectors alone, in FLUENT during stroke reversal due to the high rotational velocity. For this set of runs, which uses Eq. 4.51 for the rotational velocity, the rotational velocity becomes quite large during stroke reversal, especially for high values of h. This explanation for the discrepancy seems like a good possibility, because the difference in the negative vertical force predicted by ULVPC deviates more from the FLUENT trace at higher values of h and smaller values of c (at smaller values of c, viscous effects such as boundary layer separation are expected to be more important). Figure 4.21 shows contours of vorticity during stroke reversal for $h = 3.5$ and $c = 0.04$, which is the case that would be expected to show the most boundary layer separation. This figure shows, during the beginning of the rotational phase, the boundary layer on the top surface of the airfoil begins to grow in size for this case, and by the end of the stroke it seems to be completely separated. This separation is likely the reason why, in FLUENT, the negative vertical force peak following stroke reversal is not as large, at least for the cases with higher values of h. At lower h, it may have more

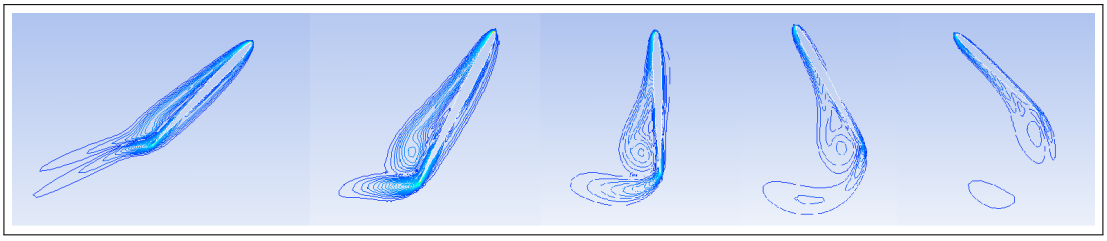


Figure 4.21: Vorticity contours during stroke reversal for $h = 3.5$, $c = 0.04$

to do with the proximity of the previously-shed wake as was the case in the first set of comparisons.

4.5 Discussion of Results

The comparisons between ULVPC and FLUENT for a 2D airfoil in hovering flight are promising. Even when the angle of attack is not kept small for the entire flapping cycle, the cycle-averaged horizontal forces predicted by ULVPC are consistent with FLUENT's, though the shape and magnitude are different. When the angle of attack is kept very small throughout the cycle, the horizontal forces match very well, even in the shape and magnitude. ULVPC tends to under-predict the vertical forces, but at the small size of an MAV, viscous effects attenuate the FLUENT vertical force peaks to be very close to the ULVPC prediction. In general, as the non-dimensional amplitude of flapping, h, increases, the vertical forces predicted by ULVPC match FLUENT's prediction more closely, suggesting that the wake interactions at low h play a significant role in the difference in vertical force predicted by the two methods. When the angle of attack is not kept low, FLUENT shows that vortex shedding occurs from the leading edge during stroke reversal. This vortex shedding reduces the amount of vertical force in the negative direction present in the FLUENT force trace versus the ULVPC force trace during the rotational phase.

Despite the differences in the force traces between the two codes, ULVPC does surprisingly well at capturing the forces produced by a 2D airfoil undergoing hovering motions, and would likely be useful to MAV designers in its current form. This is especially true if fairly high flapping amplitudes are employed. The code could be extended to three dimensions to also capture some 3D flow effects and to predict the forces on the actual wings of an MAV, instead of just a wing section.

The major differences between the force traces predicted by ULVPC and FLUENT seemed to be a result of the wake treatment and ULVPC's inability to predict vortex shedding. The latter of these problems, at least, can possibly be addressed. The leading-edge vortex is actually a well-known flow phenomenon in insect

flight and is thought to be a major contributor to force-augmentation in this low-Reynolds-number regime. It occurs due to leading-edge separation at high angles of attack and is stabilized at low local Rossby number in rotational wing motions [50]. The leading-edge vortex is discussed in detail in Chapter 2. The leading-edge vortices observed in the current study were not particularly strong, likely because even in the first motion case the angle of attack was fairly low for most of the stroke. Though separation cannot be predicted by potential flow theory, several researchers have attempted to model it by means of shed vortices from the leading edge, much like ULVPC sheds from the trailing edge [23], [34]. The most successful of these attempts was Ansari's [23], which modeled both leading-edge and trailing-edge separation for hovering flight using potential flow. With his method, good agreement was achieved against experimental data both in the force prediction and flow visualization, as can be seen in Fig. 2.1 (reprinted from Knowles, et al. [39]).

With the promising results shown by ULVPC for hovering flapping flight as a stepping stone, future work should attempt to model the leading-edge vortex in a way similar to Zbikowki's [34] (whose method also gave good agreement with experimental data for vertical force production) and Ansari's [23], which should further improve the comparison between CFD and vortex panel methods for hovering flapping flight. If such work is successful, implementation of a 3D unsteady panel code with that adaptation for hovering flight would prove particularly beneficial and insightful for designers of Micro Air Vehicles.

Chapter 5

Experimental Methods

Experimental testing is important in the design of MAV flapping wings. It can be used for a number of purposes, including validation of CFD simulations, testing of actual wings for the MAV, and testing of control systems. Experimental testing can also be used to measure real-world forces that are not present in a typical CFD simulation, such as inertial forces and vibration. With a good experimental testing setup, aerodynamic forces on flapping wings can be measured in much less time than is required to complete a CFD simulation, which can be very beneficial for MAV design.

Because there is other information in the force trace measured for flapping wings using an experimental setup, however, separating out the aerodynamic forces can be difficult. The main goal of this chapter is to investigate methods of isolating the aerodynamic forces from the other forces measured in an experimental benchtop flapping device, since this subject is not well-investigated in the literature—not to measure the forces on the actual Quad-Wing Vehicle. The benchtop system and wings are first constructed, and then forces are measured while the wings are flapping. Various methods of isolating the aerodynamic forces from the measured force trace are then investigated. The results and conclusions drawn from the experimental testing are meant to provide guidance for future engineers in developing experimental testing systems for flapping-wing MAVs.

5.1 Hardware and Test Setup

The benchtop flapping device designed in this chapter is based on the actuation system from a commercial off-the-shelf ornithopter toy called the iFly Vamp. The iFly Vamp is a small ornithopter with flapping wings made of mylar and reinforced with plastic tape. The wingspan is about 5 inches for each wing. It is designed

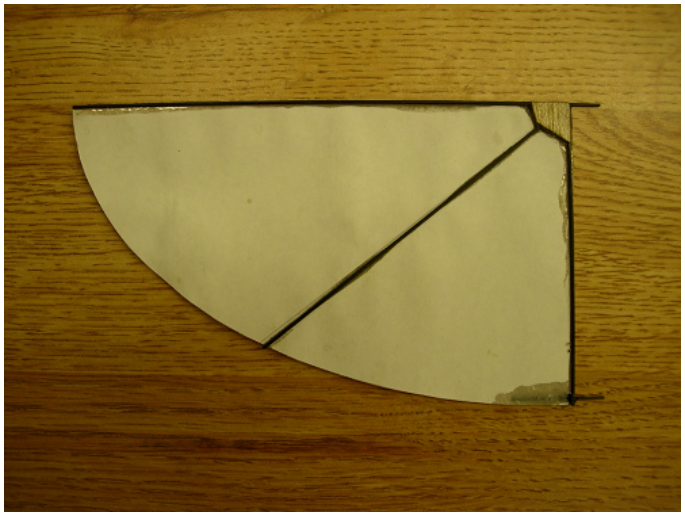


Figure 5.1: Flapping wing for experimental testing

to fly with some forward speed like a bird, not to hover like an insect, so the wings provide thrust and lift during flight. The wings flap in a vertical stroke plane during flight and there is no feathering rotation of the spar; the rotation of the wing occurs about a single axis only. However, the stock wings are very flexible, which likely helps to produce thrust by introducing a feathering motion via aeroelasticity in both the upstroke and the downstroke. The use of flexibility to aid in thrust production in this way was studied by Ho, et al. [17] and Frampton, et al. [55]. The flexibility of the iFly Vamp wings is due to their extreme light weight and lack of stiff reinforcement. Each of the stock wings weighs 0.56 g and has a quarter-ellipse planform.

The numerical studies of flapping wings in this thesis so far have dealt only with rigid wings, so the first step in designing the benchtop flapping system is to fabricate rigid wings for the tests. One of the wings used for experimental testing in this chapter is shown in Fig. 5.1, and the other wing is a mirror image of this one. The wing is designed to be much stiffer than the stock wings; it is fabricated out of paper with carbon fiber reinforcements for the leading edge, the root, and as a rib. The carbon fiber is secured to the paper with epoxy. The stiffer construction of the wing also makes it much heavier than the stock wing—each of the fabricated wings weighs 1.27 g. The wings are designed to attach to the actuator at the root leading edge and trailing edge.

5.1.1 Load Cell Selection

In addition to the wings, another critical part of the experimental flapping system is selection of a measuring device. A load cell is a sensor that produces a voltage

output signal proportional to a force input, usually by means of strain gauges inside the cell. Load cells come in many shapes, sizes, configurations, and load capacities. Many measure forces in only a single axis or even a single direction in that axis, while others have the capability to measure in two or three axes. With all these available options, selecting an appropriate load cell for a given application is not always an easy task.

Selection of the load cell for the experimental setup here is subject to a few requirements. First, the load cell must measure in both directions in at least one axis, since the wings will create positive and negative vertical forces while flapping. Second, it must be sensitive enough to resolve the relatively low-magnitude aerodynamic forces. Third, its capacity must be great enough to support the weight of the wings and the actuation system that will be placed on top of it. Finally, its size and shape must allow it to be mounted conveniently in the benchtop system.

The Futek LRF400 load cell with 0.5 lb capacity was chosen to meet these requirements. This load cell is a single-axis type that can measure in tension and compression. It is also compact and has three threaded holes on its underside that can be used for mounting. The actuation system can be attached to the receiver on top of the cell. The nonlinearity and hysteresis are rated at 0.05% of the total measuring range. The output signal is amplified to a +/- 10V scale and then converted to a digital signal by a National Instruments NI USB-6008 Data Acquisition device. National Instruments LabView software is used to convert the voltage signal to force and to record the data. The sampling frequency is 1000 Hz. The load cell requires two sets of electrical wires: two from the power supply giving 24V DC power, one which outputs the measured voltage signal, and one which is attached to ground. The amplifier uses the same 24V DC power, and it also takes in the signal from the load cell and amplifies it before sending it to the NI DAQ. The actuation system requires 2-3V DC power from the power supply. For some of the trials, the actuation system and load cell are mounted inside a vacuum chamber, and the power and signal wires are fed through the chamber. A flowchart of the experimental measurement system is shown in Fig. 5.2.

5.1.2 Design of the mounting system

The mounting system is designed with a few considerations in mind. The overall size of the system is designed to place the wings roughly in the center of the RIT wind tunnel, so that future measurements of the forces in forward-flight mode can be made. Since the load cell only measures in a single axis, it is important to allow for two different mounting setups to measure the lift force and drag or thrust

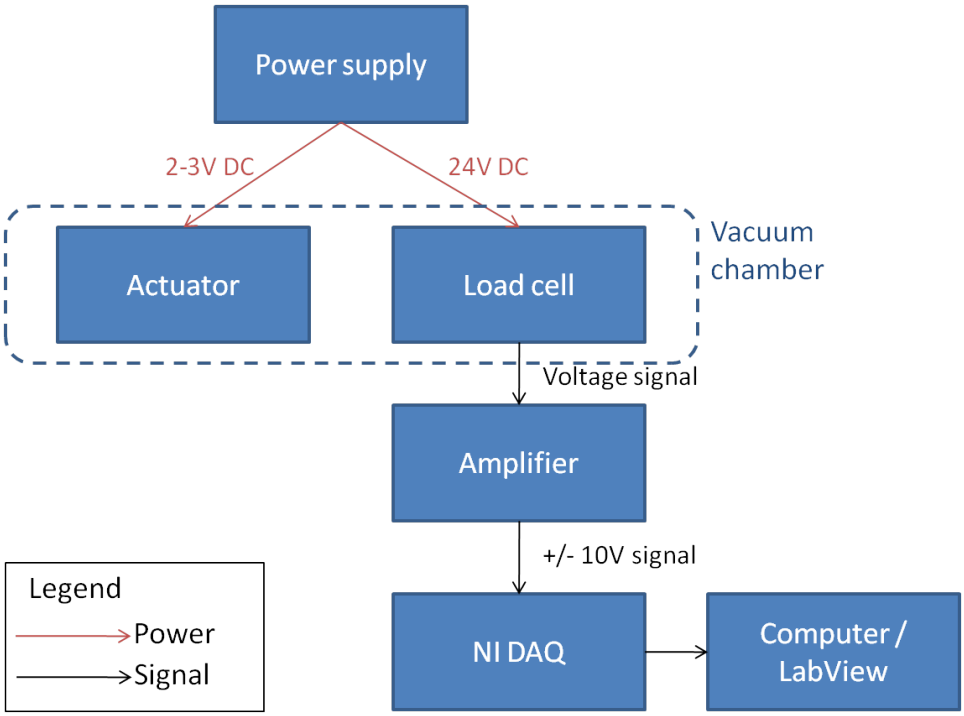


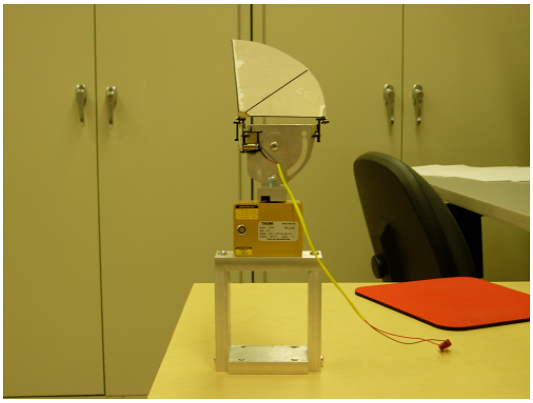
Figure 5.2: Flowchart of the experimental measurement system

separately. This requirement is accommodated by designing two different mounts, one placing the load cell's measuring axis in the vertical direction, and the other placing its measuring axis on the horizontal direction.

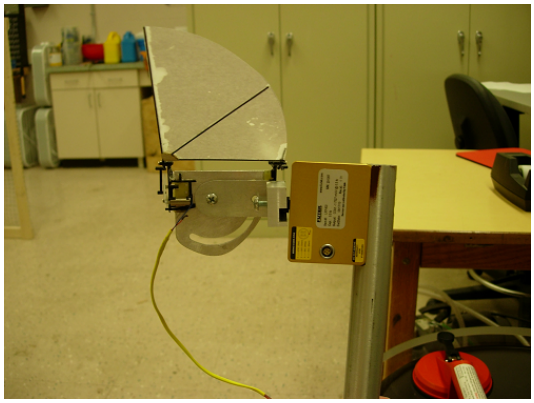
The actuation system attached to the top of the load cell rotates to allow the inclination of the stroke plane to be adjusted by up to 90°. This ability is required in order to measure lift and thrust separately, and it is also useful for hovering flight studies, because in the final design of the Quad-Wing vehicle the stroke plane may not be completely vertical. Both mounting setups were machined primarily out of aluminum and are shown in Fig. 5.3. The actuation system, including the motor, gearbox, and wing attachments from the iFly Vamp, is mounted to a thin aluminum plate that attaches to the load cell and has the ability to rotate to allow variation in the stroke plane inclination. A close-up view of the actuation system is shown in Fig. 5.4.

5.2 Isolating Aerodynamic Forces

Quite surprisingly, many of the papers found in the literature that measure the aerodynamic forces on flapping wings experimentally completely neglect to mention the inertial loads. In cases where a scaled-up version of the wing is actuated in a denser fluid, making use of dynamic scaling principles, the flapping frequency can be reduced and the magnitude of fluid forces can be increased significantly. Thus, in



(a) Lift-measuring configuration



(b) Thrust- or drag-measuring configuration

Figure 5.3: Benchtop flapping system in two different mounting configurations

such cases it is most likely reasonable to ignore the inertial loading. In fact, Sane and Dickinson performed tests to determine whether the inertial forces were important in their dynamically-scaled model of a fruit fly wing and found that they were negligible [4]. However, for normal-scale flapping models, where the flapping frequency is high and the aerodynamic forces are relatively small in magnitude, the inertial loading cannot be ignored, and correctly estimating or measuring these inertial forces is key to isolating the aerodynamic forces from the force trace. One paper in which forces on flapping wings were measured experimentally and the inertial loading was addressed was written by Ames, et al., who used a simple analytical method to estimate the inertial loads [27]. Their analytical model considered the wing as a point mass placed at the actual wing's center of mass.

Two methods of determining inertial loads are investigated here. The first uses a vacuum chamber to measure only the inertial loading at one time, which can then be subtracted off of the total measured force trace. The second method derives analytical expressions for the inertial forces acting on the wing during a flapping

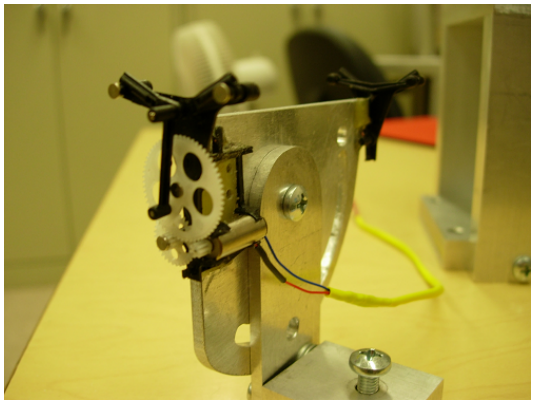


Figure 5.4: Close-up view of actuation system

cycle using first principles and applies them to the benchtop flapping device.

5.2.1 Vacuum Chamber Method

The first method of isolating the aerodynamic forces from the load cell's measurements involves a vacuum chamber. The concept here is to directly measure the inertial forces by removing the air from the system. If the flapping frequency is matched for the actuation system inside and outside of the vacuum chamber, the aerodynamic forces should be equal to the force measured while flapping in air minus the force measured while flapping in a vacuum. This approach was also taken by Bruggeman, though he was only interested in the difference in power consumption, not the difference in the force traces [28].

A commercial vacuum chamber made by LACO Technologies is used for this work. The chamber is cylindrical, is made of aluminum, and has a thick acrylic lid that is sealed by a rubber gasket. The chamber is 18 inches in outer diameter and 20 inches tall, leaving plenty of room for the wings, actuator, load cell, and mount to fit inside. It has two valves: one for attaching a vacuum pump and the other for venting. The chamber is evacuated by first attaching a vacuum pump to the main valve, then closing the vent valve, then running the pump. The main valve should be closed prior to turning off the pump. The vacuum chamber also has a vacuum pressure gauge that reads from 0 to 30 inHg, where 30 inHg is considered a full vacuum.

The vacuum chamber also is equipped with an 8-pin electrical feedthrough so that the power and signal can be passed through the chamber without releasing the vacuum. The load cell uses four wires and pins while the actuation system requires two. The connection requirements for the load cell and actuation system are described in Section 5.1 and in Fig. 5.2. All wires required to go to the load cell and actuation system are soldered with polarized connectors so that the vacuum chamber can easily be put into or taken out of the system. The outside of the vacuum chamber and the placement of the wings, actuation system, load cell, and mount, with their respective electrical connections, are shown in Figs. 5.5 and 5.6.

During the experiments, the vacuum pump was able to pull a vacuum of 29.5 inHg. In practice, a higher vacuum is desirable, but this level was sufficient to result in a significant change in the force trace measured by the load cell inside and outside of the chamber, and it also significantly changed the voltage required by the actuation system to achieve the same flapping frequency inside and outside of the chamber (in the trials presented in Section 5.3, 2.8V was required by the actuation system to flap the wings at 10.5 Hz outside of the chamber, while inside

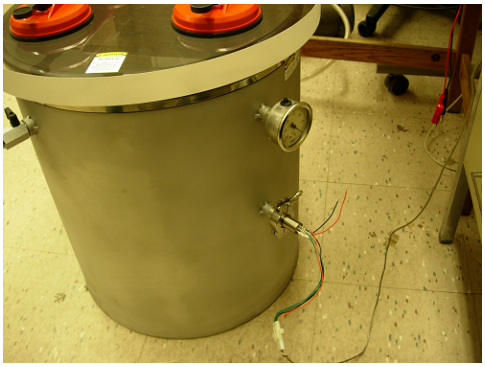


Figure 5.5: Outside view of the vacuum chamber

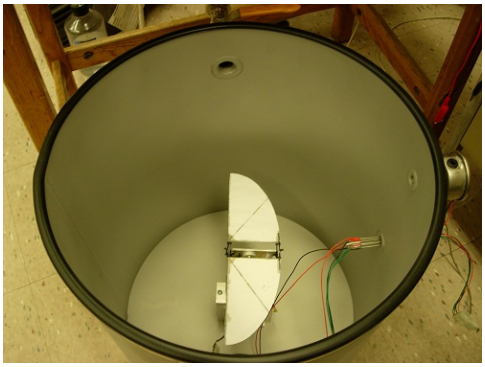


Figure 5.6: View of the system and wiring inside the vacuum chamber

the chamber only 2.0V was required for flapping at the same frequency).

5.2.2 Analytical Method

The analytical method presented here attempts to estimate the inertial forces measured by the load cell while the wing is flapping. The treatment is similar to that used by Ames, et al. [27], but considers the wing as a rigid body with a defined shape and mass distribution instead of as a point mass. The inertial loading depends on the wing's geometric and mass characteristics as well as the benchtop flapping device's kinematics. In this section, the equations to analytically determine the inertial loading for the benchtop device are developed and the kinematics parameters are simulated. With an estimate of the inertial forces measured by the load cell, the aerodynamic forces can then be estimated by subtracting the inertial forces off of the total force trace.

The inertial force acting on a flapping wing is due to the acceleration of its distributed mass. The motion of a flapping wing relative to the body is purely rotational; if the body is also accelerating then the additional inertial force due to linear acceleration can be calculated easily using Newton's Second Law of Motion. The rotational acceleration of the wing produces a moment $\vec{M}_i$ on the body given by Eq. 5.1,

$$\vec{M}_i = \mathbf{I}\dot{\vec{\Omega}} \tag{5.1}$$

where $\mathbf{I}$ is the inertia tensor of the wing and $\dot{\vec{\Omega}}$ is the rotational acceleration of the wing about its center of rotation. The force $\vec{F}_i$ is related to the moment $\vec{M}_i$ by the definition of a moment, given in Eq. 5.2,

$$\vec{M}_i = \vec{r} \times \vec{F}_i \tag{5.2}$$

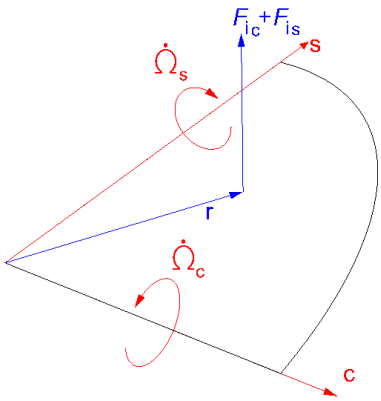


Figure 5.7: Inertial loads resulting from in-plane rotational accelerations

where $\vec{r}$ is the vector from the center of rotation to the center of mass, as shown in Fig. 5.7. Combining these two equations, the inertial forces due to angular acceleration are related to the kinematics and shape variables by Eq. 5.3.

$$\vec{r} \times \vec{F_i} = \mathbf{I}\dot{\vec{\Omega}} \tag{5.3}$$

In general, it is impossible to solve for $\vec{F_i}$ in Eq. 5.3 because there are an infinite number of force vectors that satisfy the equation, since the force vector has a magnitude and direction which are both unknown. However, if all rotations occur only about axes that are in the plane of the wing, as is the case with the benchtop flapping device (its flapping rotation occurs about the chord-axis only), then it follows that $\vec{F_i}$ acts in a direction strictly perpendicular to the wing, and Eq. 5.3 can be solved for $\vec{F_i}$. Using a property of the cross product, Eq. 5.3 can be written in the following form.

$$-\vec{F_i} \times \vec{r} = \mathbf{I}\dot{\vec{\Omega}} \tag{5.4}$$

Cross-multiplying both sides of Eq. 5.4 by $\vec{r}$ on the left results in the following relation.

$$\vec{r} \times \left(-\vec{F_i} \times \vec{r}\right) = \vec{r} \times \mathbf{I}\dot{\vec{\Omega}} \tag{5.5}$$

Using the triple product expansion, Eq. 5.5 can be expanded as follows.

$$-\vec{F_i}\left(\vec{r} \cdot \vec{r}\right) + \vec{r}\left(\vec{r} \cdot \vec{F_i}\right) = \vec{r} \times \mathbf{I}\dot{\vec{\Omega}} \tag{5.6}$$

In the above equation, $\vec{r} \cdot \vec{r}$ is equal to $|\vec{r}|^2$, and, since $\vec{r}$ and $\vec{F_i}$ are orthogonal when the rotational axes are in the plane of the wing, $\vec{r} \cdot \vec{F_i}$ is equal to 0. With these simplifications, $\vec{F_i}$ can be solved and is given by Eq. 5.7.

$$\vec{F_i} = \frac{\mathbf{I}\dot{\vec{\Omega}} \times \vec{r}}{|\vec{r}|^2} \text{ (rotations about in-plane axes only)} \tag{5.7}$$

For a flapping wing, the $\vec{r}$ vector rotates with the wing and so, as long as the $\vec{r}$ vector and the angular accelerations $\dot{\vec{\Omega}}$ are always written in the inertial frame, the contribution to inertial forces returned by Eq. 5.7 will also be correct in the inertial frame.

In order to estimate the inertial load on the benchtop flapping device, it is necessary to measure several quantities, including the mass of the wing, the center of mass of the wing, and the variation of flapping angle with time. The flapping angle is required in order to determine the $\dot{\vec{\Omega}}$. The wing designed for the benchtop test is shown in Fig. 5.1; it is a quarter-ellipse planform with dimensions roughly double the size of a typical MAV flapping wing. The mass of each wing is 1.27 g. The moment of inertia tensor $\mathbf{I}$ is calculated by assuming each wing is a flat quarter ellipse with uniformly-distributed mass. Following from this assumption, for simulation purposes the center of mass is placed at the centroid of the wing, which is 3.5 cm from the leading edge and 5.8 cm from the root. Equation 5.8 is used to determine the principle moments of inertia for a flat, two-dimensional wing rotating about an arbitrary axis. The directions c and s are in the plane of the wing, as in Fig. 5.7.

$$I_{princ} = \bar{\rho} \int_{c_1}^{c_2} \int_{s_1}^{s_2} |r\,(c,s)|^2 \, dsdc \tag{5.8}$$

In Eq. 5.8, $\bar{\rho}$ is the mass per unit wing area and r is the vector from the center of rotation to a point on the wing. The products of inertia—the terms not along the diagonal—are found using Eq. 5.9.

$$I_{prod} = \bar{\rho} \int_{c_1}^{c_2} \int_{s_1}^{s_2} (cs) \, dsdc \tag{5.9}$$

For a flat quarter-ellipse wing with root chord a and span b, these equations result in the following form for the moment of inertia tensor, arranged in the c-, s-, and n- (normal direction) axes.

$$\mathbf{I} = m \begin{bmatrix} \frac{1}{4}b^2 & -\frac{1}{2\pi}ab & 0 \\ -\frac{1}{2\pi}ab & \frac{1}{4}a^2 & 0 \\ 0 & 0 & \frac{1}{4}\left(a^2+b^2\right) \end{bmatrix} \tag{5.10}$$

In addition to the inertial loading due to angular acceleration of the wing, there is also a centripetal force due to the rotational motion. The centripetal force is

equivalent to the product of the centripetal acceleration and the mass of the wing, as given in Eq. 5.11,

$$F_c = m\frac{v_c^2}{s} \tag{5.11}$$

where F_c is the centripetal force, m is the mass of the wing, v_c is the tangential speed of a certain point on the wing, and s is the perpendicular distance from the center of flapping rotation to the point on the wing (which is in the span-axis direction). This equation describes a point mass undergoing rotation about an axis, but it can be generalized to a wing with distributed mass by differentiating and then integrating. Differentiating Eq. 5.11 with respect to s gives the following equation.

$$\frac{dF_c}{ds} = \frac{2mv_c}{s}\frac{dv_c}{ds} - \frac{mv_c^2}{s^2} + \frac{v_c^2}{s}\frac{dm}{ds} \tag{5.12}$$

Some subsitutions and simplifications can be made in Eq. 5.12. v_c itself is equal to $\dot{\phi}s$, where $\dot{\phi}$ is the instantaneous rotational flapping velocity. $\frac{dv_c}{ds}$ is then equal to $\dot{\phi}$. For a flat wing with constant mass distribution across its area, as is assumed here, m is equal to $\bar{\rho}S$, where S is the total wing area. $\frac{dm}{ds}$ can then be expressed as $\bar{\rho}\frac{\partial S}{\partial s}$, or $\bar{\rho}c\left(s\right)$, where $c\left(s\right)$ is the chord length at the spanwise position s. Using these substitutions and simplifications, the centripetal force acting on the wing is found by integrating Eq. 5.12 over the single-wing span b. The result of the simplifications and integration is Eq. 5.13.

$$F_c = \bar{\rho}\dot{\phi}^2 \int_0^b \left(S + sc\left(s\right)\right) ds \tag{5.13}$$

For a quarter-ellipse wing with span b and root chord a, the wing area S is $\pi ab/4$, and $c\left(s\right)$ is given by Eq. 5.14.

$$c\left(s\right) = \frac{a}{b}\sqrt{b^2 - s^2} \tag{5.14}$$

Substituting Eq. 5.14 into Eq. 5.13 gives the final integral used to evaluate the centripetal force on a quarter-elliptical flapping wing.

$$F_c = \bar{\rho}\dot{\phi}^2 \int_0^b \left(\frac{\pi ab}{4} + \frac{a}{b}s\sqrt{b^2 - s^2}\right) ds \tag{5.15}$$

Integration of Eq. 5.15 results in the final equation for the centripetal force acting on the wing at any time, as a function of $\dot{\phi}$. The result is given in Eq. 5.16.

$$F_c = \frac{\bar{\rho}ab^2\left(3\pi + 4\right)}{12}\dot{\phi}^2 \tag{5.16}$$

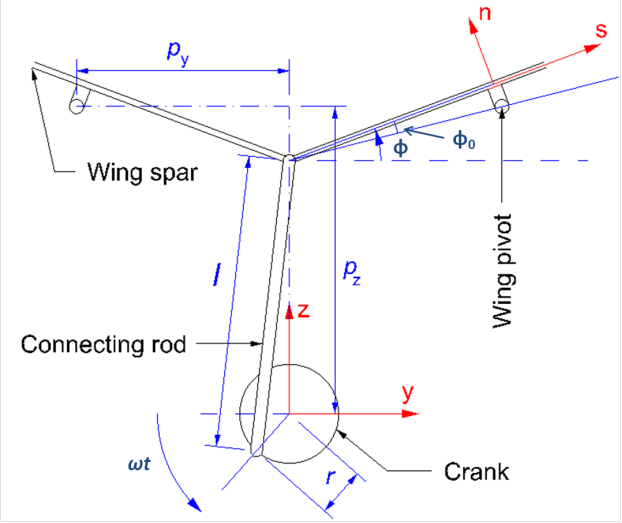


Figure 5.8: Actuator geometry from the iFly Vamp

Since Eq. 5.10 is given in terms of mass, not $\bar{\rho}$, Eq. 5.16 is rewritten below in terms of mass for consistency.

$$F_c = \frac{mb\,(3\pi + 4)}{3\pi}\dot{\phi}^2 \tag{5.17}$$

The angular speed and acceleration of the wing are more difficult to estimate than the shape and mass parameters. One way to estimate them is to use a high-speed camera and capture the flapping angle at various points in time, but that would prove difficult when flexing of the wings is taken into account, and no such camera was readily available in the lab. Another option is to simulate the flapping motion of the wings by creating a mathematical model of the system. The actuation system uses a crank, turned by the motor, with a connecting rod attaching to the ends of the wing spars. A single actuator drives both wings by moving the ends of the spars up and down vertically. Each spar rotates about a pivot outboard in the spanwise direction along the wing spar. A diagram of the system is shown in Fig. 5.8. In the diagram, black lines represent the actual geometry, blue lines and text represent physical dimensions and angles, and red lines and text show the coordinate systems of the wing and the actuator.

In the simulation of the flapping actuator, the motor speed, ω, in rad/s, is assumed to be constant. The important geometric features of the actuator were measured to determine a relationship between the crank angle ωt and the flapping angle ϕ, and are shown in Fig. 5.8. The resulting relationship is given in Eq. 5.18.

$$\phi\,(t) = \sin^{-1}\left(\frac{r\sin\,(\omega t) - \sqrt{l^2 - r^2\cos^2\,(\omega t)} + p_z}{p_y}\right) + \phi_0 \tag{5.18}$$

Taking one and two time-derivatives of Eq. 5.18 gives the flapping velocity, $\dot{\phi}(t)$, and the angular acceleration, $\ddot{\phi}(t)$. The results are given in Eqs. 5.19 and 5.20, respectively.

$$\dot{\phi}(t) = \frac{\omega r \cos(\omega t) - \frac{1}{2}\omega r^2 \sin(2\omega t)/\sqrt{l^2 - r^2\cos^2(\omega t)}}{\sqrt{p_y^2 - \left(\sqrt{l^2 - r^2\cos^2(\omega t)} - r\sin(\omega t) - p_z\right)^2}} \tag{5.19}$$

$$\ddot{\phi} = \frac{\dot{N}D - N\dot{D}}{D^2} \tag{5.20}$$

In Eq. 5.20, N is the numerator of Eq. 5.19 and D is the denominator. $\dot{N}$ and $\dot{D}$ are given by the following equations, which are then used to calculate $\ddot{\phi}$.

$$\dot{N} = \frac{1}{4}\frac{\omega^2 r^4 \sin^2(2\omega t)}{\left[l^2 - r^2\cos^2(\omega t)\right]^{3/2}} - \frac{\omega^2 r^2 \cos(2\omega t)}{\sqrt{l^2 - r^2\cos^2(\omega t)}} - \omega^2 r \sin(\omega t) \tag{5.21}$$

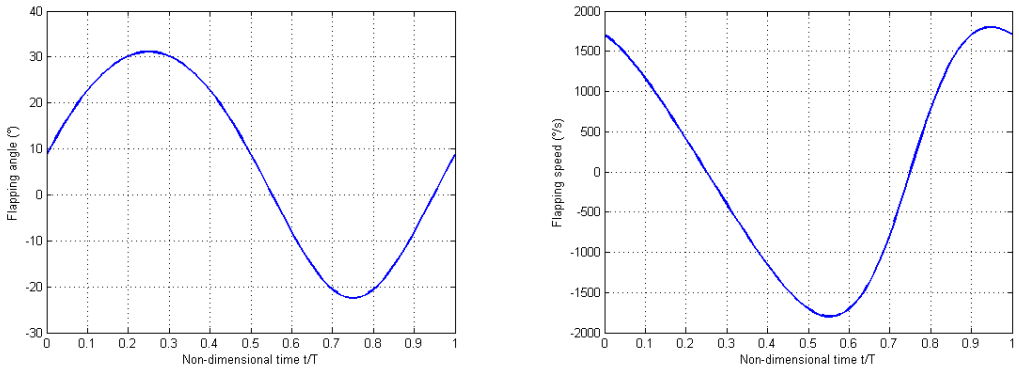
$$\dot{D} = -\frac{\left(\sqrt{l^2 - r^2\cos^2(\omega t)} - r\sin(\omega t) - p_z\right)\left(\frac{\omega r^2 \sin(2\omega t)}{2\sqrt{l^2 - r^2\cos^2(\omega t)}} - r\omega\cos(\omega t)\right)}{\sqrt{p_y^2 - \left(\sqrt{l^2 - r^2\cos^2(\omega t)} - r\sin(\omega t) - p_z\right)^2}} \tag{5.22}$$

These equations are fairly busy, but the ϕ equation works out to a modified offset sinusoid which is widened in the wing's upstroke and narrowed in the downstroke. This widening and narrowing has a more significant effect on the time derivatives, however, and the effect is especially pronounced in the angular acceleration. The flapping angle, flapping speed, and angular acceleration of the test wing resulting from Eqs. 5.18, 5.19, and 5.20 are shown in Fig. 5.9. These plots use a flapping frequency of 10.5 Hz.

With all of the kinematics and mass parameters of the benchtop flapping device measured or calculated, all inertial forces on the load cell can be estimated. The load cell is capable of measuring forces in one direction only. With the wing mounted on top of the load cell so that flapping occurs in a vertical stroke plane, as is shown in Fig. 5.3(a), the load cell measures in the z-axis. The inertial forces measured by the load cell are equal and opposite to the inertial forces acting on the wing. The z-component of the inertial forces measured by the load cell is a function of F_i, F_c, and the flapping angle of the wing, and is given by Eq. 5.23.

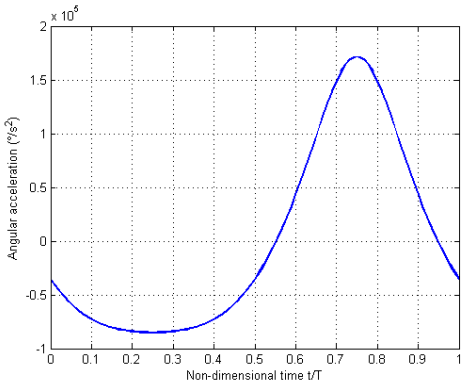
$$F_{zcell} = -F_i \cos\phi + F_c \sin\phi \tag{5.23}$$

For the benchtop flapping device used here, the flapping rotation about a single



(a) Flapping angle (°)

(b) Flapping speed ($°/s$)



(c) Angular acceleration ($°/s^2$)

Figure 5.9: Flapping angle, speed, and acceleration of the benchtop wings

axis gives an unsteady force in the y-axis and z-axis shown in Fig. 5.7, but with two wings the inertial load in the y-direction is canceled by the opposite wing. The inertial load in the z-axis, however, is doubled by the second wing.

5.3 Results of Experimental Tests

Data was recorded for the flapping wings operating at a variety of frequencies by varying the voltage to the actuation system from the power supply. Voltages ranged from 2-3V and the resulting flapping frequencies in air ranged from approximately 8-11 Hz. Note that the wings used in this experimental section are significantly larger than the nominal MAV-sized wings, due to the sizing of the motor and parts from the iFly Vamp. The flapping frequency is also fairly low compared to what would likely be seen in a normal MAV. Despite the size differences, the data from the benchtop flapping device still provides insights into experimental testing for flapping-wing MAVs. For all of the experimental testing, the load cell measured vertical force in the configuration of Fig. 5.3(a).

Figure 5.10 shows the force trace measured by the load cell over one flapping

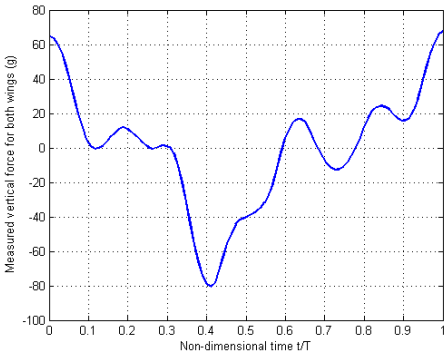
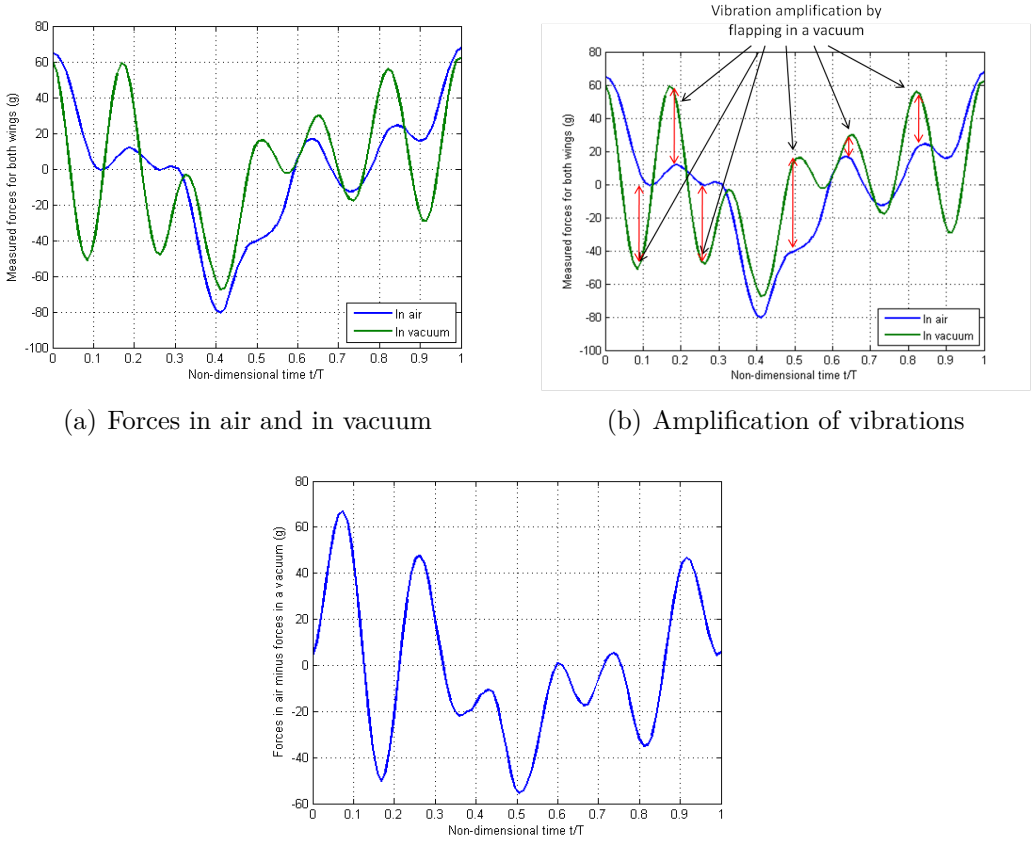


Figure 5.10: Forces measured: flapping in air at 10.5 Hz

cycle when the wings flap in air and the input voltage is 2.8V. The benchtop system has no means to mark the beginning and end of the cycle, so in this case the beginning of the cycle is taken at the point of maximum positive vertical force. The flapping frequency is estimated at 10.5 Hz for this input voltage. The maximum force is 65 g and the minimum is -81 g. The load cell has an uncertainty of about +/- 1 g. The minimum force occurs about 41% of the full flapping cycle after the maximum force. The difference in magnitude between the maximum and minimum force is likely due to the asymmetry in flapping angle, speed, and angular acceleration between the upstroke and the downstroke, which can be seen in Fig. 5.9. The force trace is also very "bumpy" throughout the cycle, likely because of vibration in the system that is transmitted to the load cell.

Force measurements in a vacuum

Following the gathering of data with the wings flapping in air, further data was gathered with the wings flapping inside the vacuum chamber to attempt to measure the inertial loads experimentally. The hope was that the vibration level would be the same, or at least roughly the same, while flapping inside a vacuum so that the subtraction of the force trace in the vacuum from the trace in air would return only the aerodynamic force. Though the chamber could not be completely evacuated with the vacuum pump available, the chamber reached a vacuum pressure of 29.5 inHg, which made a considerable difference in the power required to flap at a given frequency and also made a considerable difference in the force trace. A number of trials were run in the vacuum with the input frequency ranging from 2-3V, and the data was compared against flapping-in-air data to match the flapping frequency. It was found that, to match the frequency of 10.5 Hz that required 2.8V input in air, in the vacuum chamber only 2.0V was required. The force trace for this case is plotted against the air data and is shown in Fig. 5.11(a). The cycle-position is



(a) Forces in air and in vacuum

(b) Amplification of vibrations

(c) Force in air minus force in vacuum

Figure 5.11: Comparisons between the force traces in air and in a vacuum

matched by setting the maximum positive force at the beginning of the cycle.

As can be seen from Fig. 5.11(a), the magnitude of the maximum and minimum force are decreased by flapping in a vacuum, which would be expected since there is no aerodynamic force. The maximum force is 62 g for the vacuum force trace, while the minimum is -68 g. However, more prominent than the difference in the force magnitude is the difference in the vibration level. The vacuum force trace exhibits "bumpiness" at the same cycle locations as the force trace measured in air, but the bumps are much more extreme. This finding supports the suggestion that these bumps are due to vibration in the system, and it shows that the air likely provides considerable vibrational damping, reducing the magnitude of the vibrations. The amplification of the vibrations in the vacuum chamber are pointed out in Fig. 5.11(b).

Unfortunately, due to the vibrations in both of the force traces and the drastic difference in vibration magnitude inside and outside of the vacuum chamber, subtracting out the vacuum force trace from the air force trace does not give a good idea of the aerodynamic forces; instead, it mainly just highlights the difference in magnitude of vibrational force. This effect is shown in Fig. 5.11(c). The results

shown here do not invalidate the concept of using a vacuum to isolate the aerodynamic forces in measurement, but they do highlight the need to take very careful measures to remove vibration from the flapping system before measuring the force data. This is especially true when flapping in a vacuum, because the lack of air damping only increases the vibration level.

Analytical method of force isolation

The analytical method of estimating the inertial loads acting on the flapping wings developed in Section 5.2.2 is used to simulate the inertial loads on the benchtop device with the wings flapping at 10.5 Hz. In the simulation, the cycle starts in the middle of the upstroke where the flapping angle is ϕ_0; the middle of the downstroke occurs at t/T equal to 0.5 and the ends of each stroke occur at t/T equal to 0.25 and 0.75. The result is shown in Fig. 5.12. There are two equivalent maxima of 30 g in the analytical inertial load approximation, which occur in the later part of the upstroke and the beginning part of the downstroke, respectively. The minimum is -51 g. The asymmetry in the positive and negative forces is due to the asymmetry in the flapping angle, speed, and angular acceleration of the benchtop flapping device in the upstroke and downstroke.

As is shown in Fig. 5.11, there is a good deal of vibration in the measured force trace, even for the wings flapping in air. The initial concept of subtracting off the inertial load from the measured load is discarded for these trials, because the resulting force trace would represent not just the aerodynamic load, but the aerodynamic load plus vibration, as in Fig. 5.11(c). Instead, to get an idea of the validity of the analytical inertial-load estimate, CFD is used to estimate the aerodynamic forces, which are then added to the analytical estimate of inertial loading and the total is compared to the measured force trace.

The CFD model of the wing is identical to the CFD model used in Chapter 3, but it is scaled to match the dimensions of the experimental wing shown Fig. 5.1. The wing starts in the upstroke with the flapping angle ϕ_0 (from Eq. 5.18), which is 8.8°. ANSYS FLUENT 12.1 is used to simulate the flapping wing, and the boundary conditions and solver settings are identical to those used in Section 3.3.1 of this thesis. Only a single wing is modeled in CFD, but the vertical forces are doubled to account for the second wing in the experimental setup. The wing rotates about its chord axis in the simulation according to Eq. 5.19. The resulting aerodynamic vertical force for two wings from CFD is shown in Fig. 5.13.

The CFD force trace has a similar shape to the inertial force trace, in that in the upstroke the trace is flattened and in the downstroke it is narrowed due to the

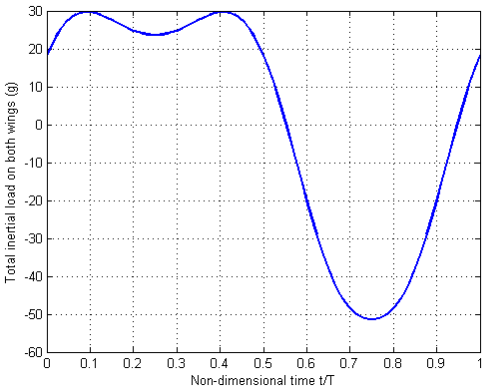


Figure 5.12: Analytical estimation of inertial loads at 10.5 Hz

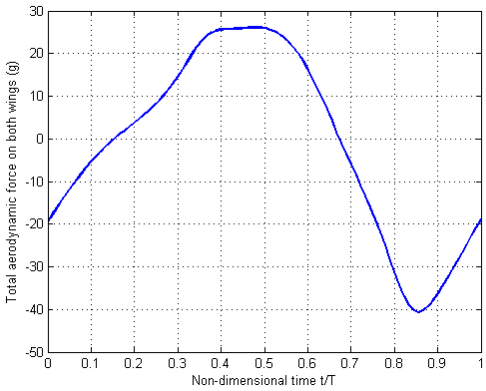


Figure 5.13: CFD estimation of aerodynamic forces at 10.5 Hz

kinematics. The maximum negative vertical force is also greater in magnitude than the maximum positive force; the maximum positive force is 26 g while the maximum negative force is -41 g. The aerodynamic force peaks also occur later in the flapping cycle than the inertial force peaks.

Addition of the analytical inertial force with the aerodynamic force simulated in CFD should give a prediction of the force measured by the load cell sans vibration. The comparison between the measured force trace in air with the wings flapping at 10.5 Hz is compared with this analytical + CFD prediction in Fig. 5.14. Note that the force trace predicted by means of the analytical expressions and CFD has been shifted in this figure so that the maxima line up with the measured force trace. Despite the vibration in the measured force trace, the comparison is remarkably close. The maximum value of the predicted force trace is 55 g, while the minimum value is -82 g. These values differ from the measured values by 15.4% and 1.2%, respectively. The reason for the greater difference in the vertical peak is not clear. Greater agreement would likely be possible if vibration in the experimental system were minimized and if the wings were completely rigid, as in the analytical and CFD treatment.

Also encouraging is the fact that, in both the experimental and predicted force trace, the negative maximum follows the positive maximum by 41% of the full cycle, instead of 50% as might be expected. The reason for this offset is because the inertial force trace has two equal positive maxima, one before the end of the upstroke and another in the beginning of the downstroke (see Fig. 5.12), and because the aerodynamic maximum and minimum values occur later in the cycle than the inertial peaks. Because of this nature, the maximum total vertical force is shifted significantly into the downstroke. The negative peak is also shifted due to the difference in cycle-position of the inertial-load minimum and the aerodynamic-

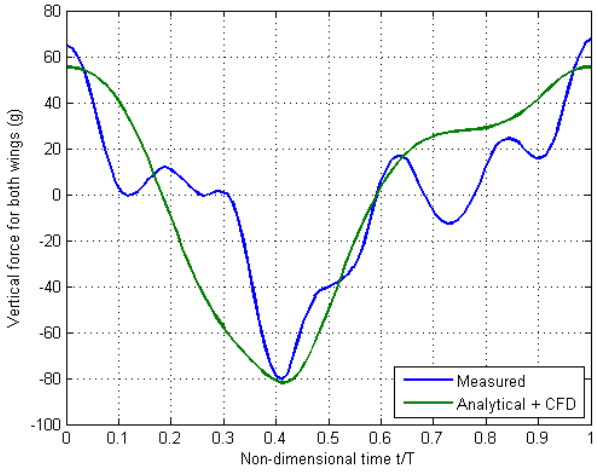


Figure 5.14: Comparison of measured load with predicted load

load minimum but not as severely because the inertial load has only a single peak in the negative direction occurring at the end of the downstroke.

The comparison between the measured force trace and the predicted force trace is encouraging; despite the vibration in the measured force trace, the analytical and computational methods are able to predict the magnitudes and cycle-positions of the force peaks as measured by the load cell (the prediction of the negative force peak is especially close). The favorable comparison suggests that analytical predictions of the inertial loads are sufficient to isolate the aerodynamic forces from measured data in flapping-wing MAVs, provided vibrations can be minimized in the system. The results also provide further validation for the FLUENT CFD model.

5.3.1 Recommendations for Experimental Testing

In this chapter, an experimental benchtop flapping device capable of measuring the forces produced in one direction by flapping wings is constructed. The purpose is to investigate means of isolating the aerodynamic forces from other forces produced during flapping, especially inertial forces, so that future researchers may be able to design benchtop devices capable of accurately measuring the aerodynamic forces produced by flapping-wing MAVs. This is a subject which is not well published in the literature.

The greatest difficulty in measuring the forces experimentally, at least for the device used here, is the prevalence of vibration appearing in the force trace. There are a few possibilities for the source of the vibration. The most likely source is the attachment between the actuation system and the load cell. The mount for the actuation system consists of an aluminum plate and receiver, which can be seen

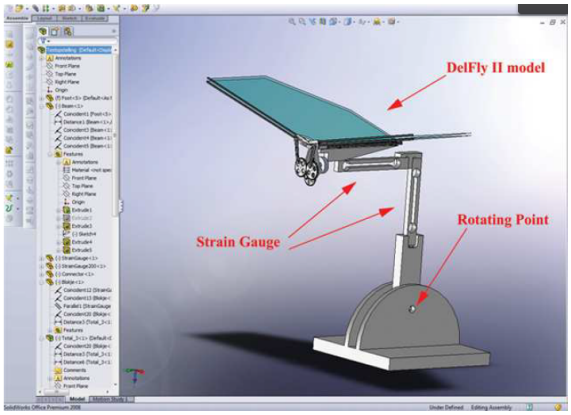


Figure 5.15: Flapping device used by Bruggeman [28]

most clearly in Fig. 5.4. The receiver has a groove (not visible in the photo) which accepts the load cell attachment and is secured with a screw. There is some play between the receiver and the load cell, and care must be taken not to tighten the screw too much because of the risk of damaging the load cell. There is also some play in the attachment between the wings and the actuator, but it is not as great and could also be present in an actual MAV.

It would be fairly difficult to improve the attachment of the actuation system to the current load cell. Instead, it may be preferable to seek a better load cell configuration. An example of a better configuration was done by Bruggeman in his analysis of the Delfly MAV [28]. In his experimental flapping device, the load cell was a dual-bending-beam type, where the beam was part of the actual mounting system. The beams were equipped with strain gauges and were designed to measure in two different axes simultaneously. The great benefit of this design is that the load cell is rigidly attached to the actuation system mount instead of relying on a screw, so there is no play in that part of the system to introduce vibrations. Bruggeman's flapping device is illustrated in Fig. 5.15 (reprinted from [28]).

The vacuum chamber testing highlights the need to minimize vibration, as the lack of air damping magnifies the vibration measured by the load cell. The vacuum chamber method may still be a useful way to isolate the aerodynamic forces, but even greater care must be used to eliminate vibrations in the benchtop flapping device when a vacuum chamber is used. Though a full vacuum was not achieved in this work, it was good enough to result in significant differences compared to flapping in air, both in the force traces and in the voltage required to attain a desired flapping frequency.

The analytical treatment used to estimate the inertial load is more encouraging. Combining a simulation of the inertial load seen by the load cell and the aerodynamic forces from a CFD simulation gave good results, though direct com-

parison with experimental data is difficult because of the vibration. Nonetheless, the predicted force peaks match the measured values well, both in magnitude and in cycle-position. This good agreement shows that analytical estimations of the inertial loading, like the ones derived in this chapter, can be quite accurate. It also provides further validation of the FLUENT CFD model. Engineers designing benchtop flapping devices for the testing of flapping MAV wings should pay careful attention to minimize vibrations, and if this is done, the aerodynamic forces can be isolated from the measured force trace by subtracting off the analytically-determined inertial loads. For best results, the analytical estimation of inertial load should treat the wing as a shape with distributed mass instead of simply as a point mass.

In addition to the methods investigated here, it is also possible to measure the aerodynamic loads with great accuracy by using dynamically-scaled models immersed in a fluid denser than air, as was done by Sane and Dickinson [4], [5] and by Lentink and Dickinson [50]. The benefit of this method is that the larger size and denser fluid increase the magnitude of fluid forces many times, which makes precise measurement easier. Of course, when using such a model, care must be taken to preserve the important non-dimensional parameters that determine the flow regime. This method also makes it possible to greatly reduce the flapping frequency, thus rendering the inertial forces and vibration negligible [4]. This seems like the best method to reliably measure the fluid forces, but it does have some downsides: it cannot be put into a wind tunnel to measure the forces in forward flight, though in theory it would be possible to build a water tunnel to serve the same purpose; it cannot be used to measure the forces on an actual MAV prototype; and this method is more complex and likely more costly. Despite these downsides, it may be the best option for many researchers. Nevertheless, this chapter has shown that the aerodynamic forces can be isolated in a normal-scale MAV benchtop flapping device if the inertial forces are estimated analytically and the vibration is minimized.

Chapter 6

Conclusions, Recommendations, and Future Work

6.1 CFD Analysis

The CFD analysis used ANSYS FLUENT 12.1 to simulate the flapping wings of the Quad-Wing Vehicle. The FLUENT solver was first validated for flapping MAV flight against data in the literature, showing excellent agreement with data from Ho, et al. [17]. The main goal of the CFD simulations was to determine optimal kinematics parameters for the wings of the Quad-Wing Vehicle in hovering mode, on the basis of vertical force production and aerodynamic power requirement.

The CFD analysis included a total of 80 single-wing simulations investigating the design space based on the kinematics parameters of flapping frequency, flapping amplitude, and feathering amplitude. The results showed that, most importantly, the maximum available flapping frequency should be used instead of the maximum available flapping amplitude in order to meet a vertical-force requirement, because this arrangement requires less power to overcome the aerodynamic moments. Additionally, for flapping with a near-vertical stroke plane like the QV does, a feathering amplitude near 90° should be used to reduce the downward vertical force and the horizontal force during the upstroke.

For "standard"-size MAV wings of single-wing span 7 cm and root chord 4 cm, flapping with a frequency of 35 Hz, a flapping amplitude of 65.4°, and a feathering amplitude of 90° represented the best case. This case met the hovering-force constraint of 35 g for all four wings (it produced 39 g) and required only 3.9 W to overcome the aerodynamic moments for all four wings. If such a high flapping frequency cannot be achieved, greater flapping amplitude can be used to meet the vertical force constraint instead, but at the expense of greater power requirement.

For example, if the flapping amplitude is increased to 90°, the flapping frequency can be decreased to 27.4 Hz to still produce 37 g of hovering force, but the aerodynamic power requirement is then 4.6 W.

The effect of scaling the wings was also investigated to provide guidance to the QV designers if the MAV is to be scaled up or down. Two additional cases were run, with shape and kinematics identical to the best case investigated in the parameter sweeps, but one with double the wing area and the other with half. The results showed that, when the wing is scaled, hovering force is proportional to S^2 (where S is the wing area) and aerodynamic power requirement is proportional to $S^{5/2}$. These relations can also be derived from the standard aerodynamic force and moment coefficient equations.

The flow field characteristics showed that, during the downstroke when the angle of attack of the wing is near 90°, there are two vortices produced on the top surface of the wing: one at the leading edge and one at the trailing edge. In the simulations performed in this chapter, the leading-edge vortex remained attached to the wing for a longer distance along the span. The leading-edge and trailing-edge vortices increase the vertical force produced during the downstroke by increasing drag and decreasing the pressure on the top surface of the wing.

Since many biological hovering fliers flap their wings in a near-horizontal instead of near-vertical stroke plane, a single simulation was performed for the same wing flapping in a horizontal stroke plane. The flapping amplitude and frequency were set identical to the best case investigated in the parameter sweeps, but the feathering motion must be symmetrical for both strokes when hovering with a horizontal stroke plane. The total feathering angle was set to an arbitrary value of 120°, so that at the midpoint of each stroke the angle of attack was 30°. The results showed that the cycle-averaged hovering force was about 25% less than the best case with a near-vertical stroke plane, but the power requirement was also about 15% less. This result shows that hovering with a horizontal stroke plane is not necessarily better than with a vertical stroke plane from a performance standpoint. However, it should be noted that the kinematics were not optimized as they were for the vertical stroke plane simulations, so better performance may in fact be possible with a horizontal stroke plane. Additionally, both the hovering force and power requirement for the case with a horizontal stroke plane exhibited a much lower degree of fluctuation compared to the vertical-stroke-plane case, despite having only slightly lower cycle-averaged values. This would be beneficial for an MAV because it would be easier on the actuation system.

Flow field results from the horizontal stroke plane simulation showed that there

is a leading-edge vortex acting on the wing during the middle of the stroke. This leading-edge vortex has been reported commonly in the literature on studies of hovering insects. The leading edge augments the lift produced at high angles of attack (in this case 30°) by creating a strong suction zone that allows the flow to reattach on the top surface of the wing instead of stalling.

Though the design of the QV calls for a near-vertical stroke plane, this research shows that it may be beneficial to investigate flapping with a horizontal stroke plane further. Beyond the possibility for greater performance, the lower force and power fluctuations during the flapping cycle are also desirable. Additionally, in the literature it has been shown that passive aeroelastic tailoring can be used to optimize the performance of flapping wings by providing feathering motion of the wing even when the actuators only allow for a single rotational degree of freedom. Aeroelastic tailoring can be used more easily to provide feathering in hover when the stroke plane is horizontal, because in that case the feathering motion should be symmetrical in both strokes.

The final CFD simulations modeled both a forewing and a hindwing with the spacing that is to be used for the QV. The results showed that the hindwing's ability to produce hovering force, as well as its aerodynamic power requirement, is decreased due to aerodynamic interaction with the forewing. The forewing's performance was very similar to the performance of the wing in the single-wing simulations, however. Three different flapping phase differences between the forewing and hindwing were tested, and all three cases exhibited this behavior, but the relative phase between the two wings did not appreciable affect the results.

6.2 Potential Flows Analysis

In the potential flows analysis, an unsteady 2D vortex panel code was developed and programmed in MATLAB. The goal of the analysis was to determine whether panel methods can be used as an alternative to CFD, which can be extremely time-consuming, for designers of hovering flapping-wing MAVs. The panel code was compared with results from FLUENT 12.1 for hovering-mode simulations.

The results showed that the panel code is able to predict the cycle-averaged horizontal forces well, though the shape and force-peak magnitude of the traces did not match the CFD result well unless the angle of attack was held small throughout the stroke. The vertical forces compared favorably with CFD in all simulations but generally underpredicted the cycle-average value. The comparison of vertical force between the two computational methods improved markedly at higher flapping

amplitudes because there was less interaction with the wake shed from previous cycles. However, the low-angle-of-attack simulations showed that high rotational rates can induced boundary layer separation even when the angle of attack is kept low; in these cases, the panel code underpredicted the cycle-averaged vertical force because the separation during rotation in CFD inhibited negative vertical force production. Quite interestingly, the vertical force comparison for the two codes did not become significantly worse when the chord was decreased to MAV-size; in fact, in some cases it improved because the vertical force peaks in CFD were attenuated by viscous effects at low Reynolds number.

The overall comparison between the panel code and CFD for hovering flapping MAV flight was quite promising, even in the current form of the code. In the literature there exist several analytical treatments of this flight regime which include modeling of the leading-edge separation that occur as well as the trailing-edge vortex shedding modeled in the panel code presented here [23], [34]. These studies have shown quite good agreement with experimental data. A similar treatment of the leading-edge separation and leading-edge vortex could be done in a panel code as well, which should improve usefulness of the code. If such modeling is successful, the panel code could be extended to three dimensions to simulate actual flapping wings of a hovering MAV, instead of just simulating 2D airfoils, and could prove to be a very valuable tool for MAV designers considering its significant time-savings compared to CFD. Additionally, aeroelasticity could be modeled in a panel code by coupling it with the governing structural equations.

6.3 Experimental Analysis

The experimental analysis presented in this thesis focused on isolating aerodynamic forces from a measured force trace. Measured forces from a benchtop flapping device include contributions from the aerodynamics, the inertial loading, and vibration. Using the results from this research, later researchers will have a better idea of how to create a successful experimental flapping device to measure aerodynamic forces for MAVs. Isolation of the aerodynamic loads in experimental flapping devices has not been well-studied in the literature.

Two different methods were used in this thesis to attempt to isolate the aerodynamic forces. The first was the vacuum-chamber method. In this method, the forces on the flapping wings were measured both inside and outside of the vacuum chamber. The aerodynamic force should be the difference between the force measured outside the chamber and inside the chamber. The results from the vacuum chamber

testing, however, were not useful for isolating the aerodynamic force, because there was an excessive amount of vibration visible in the force trace. In fact, both the force traces inside and outside of the chamber exhibited a significant amount of vibration, but it was amplified greatly when flapping in a vacuum. This result shows that the air actually provides a significant amount of vibrational damping. Though the vibration made it impossible to isolate the aerodynamic force by subtracting the vacuum force trace from the in-air trace, it did highlight the points in the cycle where vibration played a significant role. In future tests, the vacuum-chamber method may still prove useful for isolating the aerodynamic forces if the vibration in the system can be minimized.

The second method of isolating the aerodynamic force used an analytical estimation of the inertial loading. The method followed a similar approach as was used by Ames, et al. [27], but built on it; their method treated the wing as a point mass, but the analytical method presented here treated the wing as a flat shape with a distributed mass. The inertial load is made up of contributions from the angular acceleration and the centripetal acceleration. Since the vacuum-chamber method showed that there was vibration at certain locations in the measured force trace, in this case the aerodynamic forces were not determined by subtracting them off of the measured trace. Instead, a CFD simulation of the experimental flapping device was performed to predict the aerodynamic loads, and these were added to the analytical prediction of the inertial loads and the sum was compared to the measured force trace. The results were surprisingly good, as the magnitudes and relative cycle-positions of the stroke peaks matched quite well between the predicted and measured forces. This comparison showed that the analytical prediction of inertial forces and the CFD simulation of aerodynamic forces were both quite good. The analytical method developed here could be used in future experimental studies to isolate the aerodynamic forces from a measured force trace, provided vibration in the experimental flapping device can be minimized.

The large amount of vibration in the experimental flapping device is likely due mostly to the attachment between the load cell and the actuation mount, which uses a screw and has some play. In future benchtop flapping devices, it would be better to use a rigid connection between the two components. One example of such a connection was developed by Bruggeman [28], whose measuring device used strain gauges mounted to a cantilevered beam setup to measure two components of force independently. In his setup, the beams were part of the mount for the actuation system, so there was no play between the measuring device and the actuation system. Future researchers should use a setup like that one or some other

setup that minimizes vibration. With the vibration playing only a minor part in the force trace, the two methods presented in Chapter 5 could be used to isolate the aerodynamic force. Another option is to use a dynamically-scaled wing operating at lower frequency in a denser fluid, as was done by Sane and Dickinson [4], [5] and Lentink and Dickinson [50]. This method increases the magnitude of fluid forces and decreases the importance of inertial loads and vibration, but it requires the important non-dimensional parameters to be carefully matched.

Bibliography

[1] L. Petricca, P. Ohlckers, and C. Grinde, "Micro- and nano-air vehicles: State of the art," *International Journal of Aerospace Engineering*, vol. 2011, no. 1, 2011.

[2] J. Ratti and G. Vachtsevanos, "Towards energy efficiency in micro hovering air vehicles," in *IEEE 2011 Aerospace Conference*, Institute of Electrical and Electronics Engineers, 2011.

[3] R. J. Bomphrey, G. K. Taylor, and A. L. Thomas, "Smoke visualization of free-flying bumblebees indicates independent leading-edge vortices on each wing pair," *Experiments in Fluids*, vol. 46, pp. 811–821, 2009.

[4] S. P. Sane and M. H. Dickinson, "The control of flight force by a flapping wing: Lift and drag production," *The Journal of Experimental Biology*, no. 204, pp. 2607–2626, 2001.

[5] S. P. Sane and M. H. Dickinson, "The aerodynamic effects of wing rotation and a revised quasi-steady model of flapping flight," *The Journal of Experimental Biology*, no. 205, pp. 1087–1096, 2002.

[6] H. Liu and K. Kawachi, "Leading-edge vortices of flapping and rotary wings at low reynolds number," in *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications* (T. J. Mueller, ed.), American Institute of Aeronautics and Astronautics, 2001.

[7] D. J. Pines and F. Bohorquez, "Challenges facing future micro-air-vehicle development," *Journal of Aircraft*, vol. 43, no. 2, pp. 290–305, 2006.

[8] J. M. Grasmeyer and M. T. Keennon, "Development of the black widow micro air vehicle." AIAA-2001-0127, 2001.

[9] G. de Croon, K. de Clercq, R. Ruijsink, B. Remes, and C. de Wagter, "Design, aerodynamics, and vision-based control of the delfly," *International Journal of Micro Air Vehicles*, vol. 1, no. 2, 2009.

[10] K. D. Jones, C. J. Bradshaw, J. Papadopoulos, and M. F. Platzer, "Bio-inspired design of flapping-wing micro air vehicles," *The Aeronautical Journal*, vol. 109, pp. 385–393, 2005.

[11] "Micro helicopters and ornithopters." http://www.proxdynamics.com/r_d/micro_helicopters_and_ornithopters/, 2011.

[12] "Nano hummingbird." http://www.avinc.com/nano/, 2011.

[13] J. Ratti, R. Goely, S.-J. Kimz, J.-H. Moonx, T. D. Pappasy, G. Vachtsevanos, and M. Roemer, "Bio-inspired micro-air vehicle: Design and control issues," in *AIAA Infotech Conference*, American Institute of Aeronautics and Astronautics, 2010.

[14] J. Ratti and G. Vachtsevanos, "High endurance, micro aerial surveillance and reconnaisance robot," in *2011 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, Institute of Electrical and Electronics Engineers, 2011.

[15] J. Ratti, E. M. Jones, and G. Vachtsevanos, "Fixed frequency variable amplitude (fifva) actuation systems for micro aerial vehicles," in *IEEE International Conference on Robotics and Automation*, Institute of Electrical and Electronics Engineers, 2011.

[16] D. L. Raney and E. C. Slominski, "Mechanization and control concepts for biologically inspired micro air vehicles," *Journal of Aircraft*, vol. 41, pp. 1257–1265, 2004.

[17] S. Ho, H. Nassef, N. Pornsinsirirak, Y.-C. Tai, and C.-M. Ho, "Unsteady aerodynamics and flow control for flapping wing flyers," *Progress in Aerospace Sciences*, vol. 39, pp. 635–681, 2003.

[18] C. Ellington, "The novel aerodynamics of insect flight: Applications to micro-air vehicles," *Journal of Experimental Biology*, vol. 202, pp. 3439–3448, 1999.

[19] A. Azuma, M. Okamoto, and K. Yasuda, "Aerodynamic characteristics of wings at low reynolds number," in *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications* (T. J. Mueller, ed.), American Institute of Aeronautics and Astronautics, 2001.

[20] R. Madangopal, Z. A. Khan, and S. K. Agrawal, "Biologically inspired design of small flapping wing air vehicles using four-bar mechanisms and quasi-steady aerodynamics," *Journal of Mechanical Design*, vol. 127, pp. 809–816, 2005.

[21] S. P. Sane, "Review: The aerodynamics of insect flight," *Journal of Experimental Biology*, vol. 206, pp. 4191–4208, 2003.

[22] T. Theodorsen, "General theory of aerodynamic instability and the mechanism of flutter," Tech. Rep. 496, National Advisory Committee for Aeronautics, 1935.

[23] S. Ansari, *A nonlinear, unsteady, aerodynamic model for insect-like flapping wings in the hover with micro air vehicle applications*. PhD thesis, Cranfield University (RMCS Shrivenham), 2004.

[24] K. C. Hall and S. R. Hall, "A rational engineering analysis of the efficiency of flapping flight," in *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications* (T. J. Mueller, ed.), American Institute of Aeronautics and Astronautics, 2001.

[25] K.D.Jones, B. Castro, O. Mahmoud, S. Pollard, and M. Platzer, "A collaborative numerical and computational investigation of flapping-wing propulsion," in *40th Aerospace Sciences Meeting & Exhibit*, American Institute of Aeronautics and Astronautics, 2002.

[26] M. J. C. Smith, P. J. Wilkin, and M. H. Williams, "The advantages of an unsteady panel method in modeling the aerodynamic forces on rigid flapping wings," *The Journal of Experimental Biology*, vol. 1993, pp. 1073–1083, 1996.

[27] R. Ames, O. Wong, and N. Komerath, "On the flowfield and forces generated by a flapping rectangular wing at low reynolds number," in *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications* (T. J. Mueller, ed.), American Institute of Aeronautics and Astronautics, 2001.

[28] B. Bruggeman, "Improving flight performance of delfly ii in hover by improving wing design and driving mechanism," Master's thesis, Delft University of Technology, October 2010.

[29] R. F. Larijani and J. D. DeLaurier, "A nonlinear aeroelastic model for the study of flapping wing flight," in *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications* (T. J. Mueller, ed.), American Institute of Aeronautics and Astronautics, 2001.

[30] I. Garrick, "Propulsion of a flapping and oscillating airfoil," Tech. Rep. 567, National Advisory Committee for Aeronautics, 1936.

[31] T. von Kármán and J. Burgers, *Aerodynamic Theory*, vol. II, ch. General Aerodynamic Theory—Perfect Fluids. Julius Springer, 1935.

[32] K. Jones and M. Platzer, "Numerical computation of flapping-wing propulsion and power extraction," in *35th Aerospace Sciences Meeting & Exhibit*, American Institute of Aeronautics and Astronautics, 1997.

[33] T. von Kármán and W.R.Sears, "Airfoil theory for non-uniform motion," *Journal of Aeronautical Sciences*, vol. 8, no. 3, pp. 104–108, 1938.

[34] R. Zbikowski, "On aerodynamic modelling of an insect-like flapping wing in hover for micro air vehicles," *Philosophical Transactions of the Royal Society*, vol. 360, pp. 273–290, 2002.

[35] C. Pedersen, *An indicial-Polhamus model of aerodynamics of insect-like flapping wings in hover*. PhD thesis, Cranfield University (RMCS Shrivenham), 2003.

[36] F. Minotti, "Unsteady two-dimensional theory of a flapping wing," *Physical Review E*, vol. 66, pp. 1–10, 2002.

[37] M. H. Dickinson and K. G. Götz, "Unsteady aerodynamic performance of model wings at low reynolds number," *Journal of Experimental Biology*, vol. 174, pp. 45–64, 1993.

[38] M. H. Dickinson, F.-O. Lehmann, and S. P. Sane, "Wing rotation and the aerodynamic basis of insect flight," *Science*, vol. 284, pp. 1954–1960, 1999.

[39] K. Knowles, S. A. Ansari, P. C. Wilkins, and R. W. Zbikowski, "Recent progress towards developing an insect-inspired flapping-wing micro air vehicle," in *Platform Innovations and System Integration for Unmanned Air, Land and Sea Vehicles*, NATO Research and Technology Organisation, 2007.

[40] Y. Liu and M. Sun, "Wing kinematics measurement and aerodynamics of hovering droneflies," *Journal of Experimental Biology*, vol. 211, pp. 2014–2025, 2008.

[41] M. Sun and J. Tang, "Unsteady aerodynamic force generation by a model fruit fly wing in flapping motion," *The Journal of Experimental Biology*, vol. 205, pp. 55–70, 2002.

[42] C. P. Ellington, V. D. Berg, C. Willmott, and A. L. R. Thomas, "Leading-edge vortices in insect flight," *Nature*, vol. 384, pp. 626–630, 1996.

[43] J. D. Anderson, *Fundamentals of Aerodynamics*. New York: McGraw-Hill Higher Education, 3 ed., 2001.

[44] J. H. Wu, Y. L. Zhang, and M. Sun, "Hovering of model insects: simulation by coupling equations of motion with navier stokes equations," *Journal of Experimental Biology*, vol. 212, pp. 3313–3329, 2009.

[45] M. Ashraf, J. C. Lai, and J. Young, "Numerical analysis of flapping wing aerodynamics," in *16th Australian Fluid Mechanics Conference*, vol. 195, American Institute of Aeronautics and Astronautics, 2007.

[46] K.D.Jones, B.M.Castro, O.Mahmoud, and M.F.Platzer, "A numerical and experimental investigation of flapping-wing propulsion in ground effect," in *40th Aerospace Sciences Meeting & Exhibit*, American Institute of Aeronautics and Astronautics, 2002.

[47] K. Jones, C. Dohring, and M. Platzer, "Wake structures behind plunging airfoils: A comparison of numerical and experimental results," in *34th Aerospace*

Sciences Meeting & Exhibit, American Institute of Aeronautics and Astronautics, 1996.

[48] C. Ellington, "The aerodynamics of hovering insect flight. i. the quasi-steady analysis.," *Philosophical Transactions of the Royal Society of London*, vol. 305, pp. 1–15, 1984.

[49] T. Weis-Fogh, "Quick estimates of fitness in hovering animals, including novel mechanisms for lift production," *Journal of Experimental Biology*, vol. 79, pp. 169–230, 1973.

[50] D. Lentink and M. H. Dickinson, "Rotational accelerations stabilize leading edge vortices on revolving fly wings," *Journal of Experimental Biology*, vol. 212, pp. 2705–2719, 2009.

[51] D. Greenwell, "Simple engineering model for delta-wing vortex breakdown," *Journal of Aircraft*, vol. 40, pp. 402–405, 2002.

[52] T. Y. Hubel and C. Tropea, "The importance of leading edge vortices under simplified flapping flight conditions at the size scale of birds," *Journal of Experimental Biology*, vol. 213, pp. 1930–1939, 2010.

[53] W. J. Maybury and F.-O. Lehmann, "The fluid dynamics of flight control by kinematic phase lag variation between two robotic insect wings," *Journal of Experimental Biology*, vol. 207, pp. 4707–4726, 2004.

[54] K. Jones, T. Lund, and M. Platzer, "Experimental and computational investigation of flapping wing propulsions for micro air vehicles," in *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications* (T. J. Mueller, ed.), American Institute of Aeronautics and Astronautics, 2001.

[55] K. D. Frampton, M. Goldfarb, D. Monopoli, and D. Cveticanin, "Passive aeroelastic tailoring for optimal flapping wings," in *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications* (T. J. Mueller, ed.), American Institute of Aeronautics and Astronautics, 2001.

[56] ANSYS, Inc., *ANSYS FLUENT 12.0: Theory Guide*.

[57] Z. J. Wang, "The role of drag in insect hovering," *Journal of Experimental Biology*, vol. 207, pp. 4147–4155, 2004.

[58] J. Katz and A. Plotkin, *Low-Speed Aerodynamics: From Wing Theory to Panel Methods*. McGraw-Hill Series in Aeronautical and Aerospace Engineering, New York: McGraw-Hill, 1991.

[59] R. W. Fox, A. T. McDonald, and P. J. Pritchard, *Introduction to Fluid Mechanics*. John Wiley & Sons, Inc., 6 ed., 2004.

[60] C. Ellington, "The aerodynamics of hovering insect flight. vi. lift and power requirements.," *Philosophical Transactions of the Royal Society of London*, vol. 305, pp. 145–181, 1984.

Appendix A

Data used to create surface plots in Chapter 3

In the following tables, flapping amplitude varies by column and flapping frequency varies by row.

Table A.1: Usable vertical force (g)

—	20°	30°	40°	50°	60°	70°	80°	90°
5 Hz	0.125	0.209	0.314	0.441	0.601	0.777	0.950	1.12
10 Hz	0.515	0.854	1.28	1.79	2.44	3.14	3.84	4.54
15 Hz	1.17	1.94	2.89	4.07	5.54	7.11	8.71	10.3
20 Hz	2.10	3.47	5.16	7.26	9.88	12.7	15.5	18.3
25 Hz	3.29	5.44	8.09	11.4	15.5	19.8	24.3	28.8
30 Hz	4.76	7.85	11.7	16.4	22.3	28.6	35.1	41.5
35 Hz	6.49	10.7	15.9	22.3	30.4	38.9	47.7	56.6

Table A.2: Aerodynamic power requirement (W)

—	20°	30°	40°	50°	60°	70°	80°	90°
5 Hz	0.0011	0.0021	0.0038	0.0067	0.0111	0.0172	0.0249	0.0347
10 Hz	0.0093	0.0171	0.0310	0.0537	0.0894	0.138	0.200	0.279
15 Hz	0.0314	0.0578	0.105	0.181	0.303	0.466	0.677	0.947
20 Hz	0.0746	0.137	0.249	0.431	0.716	1.10	1.60	2.25
25 Hz	0.146	0.268	0.486	0.841	1.40	2.15	3.14	4.41
30 Hz	0.252	0.464	0.840	1.45	2.42	3.73	5.44	7.62
35 Hz	0.400	0.737	1.33	2.31	3.84	5.92	8.62	12.1

Appendix B

Unsteady Linear Vortex Panel Code (ULVPC) for MATLAB

B.1 ULVPC main function

```
function [results_u,forces,t_hist,CPUtime]= ...
ULVPC(body,upstream,pitch_init,kin_func,delt,nsteps)
%Unsteady Linear Vortex Panel Code (ULVPC) for airfoils
%The airfoil's center of rotation is initially at (0,0) and moves
%   according to the velocity functions defined in 'kin_func'
%Input 'body' is an n+1 x 2 matrix that defines the airfoil shape,
%   where n is the number of surface panels.  Col 1 is the vector
%   of x-values and Col 2 is the vector of z-values.
    %body must be defined clockwise beginning and ending at the
    %   trailing edge and must have a sharp trailing edge.
    %The chordline is expected to be horizontal with the desired
    %   center of rotation placed at the origin (0,0)
%Input 'upstream' is a vector containing the ambient pressure and
%   density, in the form [P_inf rho_inf]
%Input 'pitch_init' is the initial pitch angle of the airfoil
%   (rotation about y-axis), in degrees.  The initial location of
%   the airfoil's  of rotation is assumed to be at (x,z)=(0,0).
%Input 'kin_func' is a function that returns [U W thetadot] with
%   time as an input
%Input 'delt' is the time step size (assumed constant)
%Input 'nsteps' is the total number of time steps, so that end time
%   is nsteps*delt
%Output 'results_u' is an array of structures which contain results
%   from each time step.
%Output 'forces' is an array containing time histories for lift,
```

```
%   drag, and moment.  The size is nsteps+1 x 3.
%Output 't_hist' stores the flow time at each iteration.  The size
%   is nsteps+1 x 1.
%Output 'CPUtime' is the total computing time.

tic;
%% Preliminary Computations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Discretize geometry and compute influence coeffecients of the
%   airfoil's bound circulation distribution on the surface collo-
%   cation points
[geometry_init,coef_init]=init_comp(body);
n=geometry_init.n;                  %count of airfoil panels
Avorpan=coef_init.Avorpan;          %Influence coefficients of
                                    %   vortex panels

results_u(nsteps+1).geometry=[];    %initialize results storage
                                    %   array of structures
forces=zeros(nsteps+1,3);           %initialize forces and moments
                                    %   array
t_hist=zeros(nsteps+1,1);           %initialize time array
time=0;                             %set initial time to 0
stepnum=0;                          %time step counter
if delt<=0
    error('Time step must be positive.')
end
if nsteps<=0
    error('Number of steps must be a positive integer.')
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Time Loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while stepnum<=nsteps
    %Write progress
    fprintf('Percent complete: %2.2f  Time step: %u\n', ...
    stepnum^2/nsteps^2*100,stepnum);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Update position, orientation, and of airfoil velocity of the
    %   airfoil surface at the current time step, and adjust
    %   influence coefficients as necessary for current orientation

    %read inputs from previous time step
    if time==0
```

```
    vel0=[0 0 1/delt*pitch_init*pi/180];    %airfoil will be
                                            %   rotated to match
                                            %   desired initial
                                            %   pitch angle
    loc0=[0 0 0];
else
    loc0=results_u(stepnum).loc;
    vel0=results_u(stepnum).vel;
end
t0=time-delt;

%perform update
[loc1,vel1,geometry1,coef1]=update_position ...
(kin_func,geometry_init,coef_init,loc0,vel0,t0,delt);
geometry1.S=geometry_init.S;
geometry1.Sp=geometry_init.Sp;
geometry1.chord=geometry_init.chord;

%place outputs into data structure for current time step
results_u(stepnum+1).geometry=geometry1;
results_u(stepnum+1).loc=loc1;
results_u(stepnum+1).vel=vel1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Compute location of shed vortex by the airfoil between this
%   time step and the previous one
    if time~=0
        TEold=results_u(stepnum).geometry.TE;   %old trailing
                                                %   edge location
        TEnew=geometry1.TE;                     %new trailing
                                                %   edge location
        %place vortex along the path taken by the trailing edge
        xwnew=0.3*TEold(1)+0.7*TEnew(1);
        zwnew=0.3*TEold(2)+0.7*TEnew(2);
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Compute A, B, C matrices for this time step
%read inputs
if time==0
    wake.present=0;
    prev_boundcirc=0;
else
```

```
    if stepnum>1    %Previous wake does not exist for stepnum=2
        %Augment previous wake with new shed vortex
        wake=results_u(stepnum).wake;
        wake.xw=[wake.xw;xwnew];
        wake.zw=[wake.zw;zwnew];
    else
        wake.xw=xwnew;
        wake.zw=zwnew;
        wake.Gamw=[];
    end
    wake.present=1;
    prev_boundcirc=results_u(stepnum).surface.bound_circ;
end

%perform computations
[~,B,~,wakenew,boundcirc]= ...
matrices(geometry1,vel1,Avorpan,wake,prev_boundcirc);

%write outputs as appropriate
results_u(stepnum+1).surface.gamma=B(1:n+1);
results_u(stepnum+1).surface.bound_circ=boundcirc;
coef1.g=coef_init.g;
coef1.h=coef_init.h;
if time~=0
    coef1.e=wakenew.coef.e;
    coef1.f=wakenew.coef.f;
    coef1.theta=wakenew.coef.theta;
    wake.Gamw=wakenew.Gamw;
end
results_u(stepnum+1).coef=coef1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Compute pressure distribution on airfoil, lift, drag, moment,
%   and coefficients
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%read inputs
coef1.curr=coef1;
if stepnum==0;
    %No information for previous velocity potentials
    gamma1=results_u(stepnum+1).surface.gamma;
    Gamw1=0;
elseif stepnum==1
    %Information for velocity potential from previous time step
    %   only
    gamma1=[results_u(stepnum+1).surface.gamma, ...
```

```
        results_u(stepnum).surface.gamma];
        coef1.old=results_u(stepnum).coef;
        Gamw1=wakenew.Gamw;
    else
        gamma1=[results_u(stepnum+1).surface.gamma, ...
        results_u(stepnum).surface.gamma];
        coef1.old=results_u(stepnum).coef;
        Gamw1=[wakenew.Gamw [results_u(stepnum).wake.Gamw;0]];
    end

    %perform computations
    [press,gradphi,dphi,forci]=press_comp ...
    (stepnum,coef1,gamma1,Gamw1,upstream,vel1,geometry1,delt);

    %write outputs
    results_u(stepnum+1).surface.P=press(:,1);
    results_u(stepnum+1).surface.Pgage=press(:,2);
    results_u(stepnum+1).surface.V=gradphi;
    results_u(stepnum+1).surface.dphi=dphi;
    forces(stepnum+1,:)=forci;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Compute wake roll-up effect
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if time==0
        results_u(stepnum+1).wake='Not present';
    else
        %compute wake roll-up
        newwake=wake_rollup(geometry1,gamma1,wake,delt);

        %update wake locations
        wake.xw=newwake.xw;
        wake.zw=newwake.zw;
        %write results
        results_u(stepnum+1).wake=wake;
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Step forward in time
    t_hist(stepnum+1)=time;
    time=time+delt;     %Increment time
    stepnum=stepnum+1;  %Increment time step counter
    clc
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
CPUtime=toc;
```

B.2 Second-level functions

B.2.1 Initial calculations

```
function [geometry,coef]=init_comp(body)
%Function to compute initial data required for unsteady panel code
%Input 'body' is an n+1 x 2 matrix that defines the airfoil shape,
%   where n is the number of surface panels.  Col 1 is the vector of
%   x-values and Col 2 is the vector of z-values.
    %body must be defined clockwise beginning and ending at the
    %   trailing edge and must have a sharp trailing edge.
    %The chordline is expected to be horizontal with the desired
    %   center of rotation placed at the origin (0,0)
%Output 'geometry' is a data structure consisting of the following
%   fields:
    %geometry.n: number of panels
    %geometry.X: x-values of body vertices
    %geometry.Z: z-values of body vertices
    %geometry.TE: (x,z) pair representing the trailing edge location
    %geometry.x: x-values of collocation points
    %geometry.z: z-values of collocation points
    %geometry.rc: vectors from center of rotation to collocation
    %   points
    %geometry.chord: chord length
    %geometry.qchord: location of the quarter-chord
    %geometry.al: relative panel angles
    %geometry.norm: panel normal vectors
    %geometry.S: panel lengths
    %geometry.Sp: panel lengths to collocation point
%Output 'coef' is a data structure consisting of the following
%   fields:
    %coef.a: coefficients of induced velocity in x-direction due to
    %   contribution from first panel endpoints
    %coef.b: coefficients of induced velocity in z-direction due to
    %   contribution from first panel endpoints
    %coef.c: coefficients of induced velocity in x-direction due to
    %   contribution from second panel endpoints
    %coef.d: coefficients of induced velocity in z-direction due to
    %   contribution from second panel endpoints
    %coef.g: coefficients of velocity potential due to contribution
```

```
    %   from first panel endpoints
    %coef.h: coefficients of velocity potential due to contribution
    %   from second panel endpoints
    %coef.Avorpan: upper-left portion of A matrix due to surface
    %   circulation distribution
%Note that all quantities except for n, S, Sp, and Avorpan must be
%   transformed in accordance with the current position and
%   orientation of the airfoil

n=size(body,1)-1;                               %number of body panels
X=body(:,1);                                    %panel endpoints (x)
Z=body(:,2);                                    %panel endpoints (z)
TE=[X(1) Z(1)];                                 %trailing edge location
chord=max(body(:,1))-min(body(:,1));            %airfoil chord
qchord=[min(body(:,1))+chord/4 0];              %quarter-chord location
x=zeros(n,1);                                   %collocation points (x)
z=zeros(n,1);                                   %collocation points (z)
al=zeros(n,1);                                  %panel angles
norm=zeros(n,2);                                %panel normal directions
S=zeros(n,1);                                   %panel lengths
Sp=zeros(n,1);                                  %panel lengths to
                                                %   collocation points

%Calculate panel angles, normal vectors, tangential vectors, panel
%   lengths, and place collocation points on panels
for i=1:n
    %Calculate panel angle
    al(i)=atan2(Z(i)-Z(i+1),X(i+1)-X(i));

    %Calculate normal vectors
    norm(i,:)=[sin(al(i)) cos(al(i))];

    %Calculate panel lengths and lengths to collocation points
    S(i)=sqrt((X(i+1)-X(i))^2+(Z(i+1)-Z(i))^2);
    if i==1             %collocation points on first and last panels
                        %   are moved closer to the TE
        Sp(i)=S(i)/4;
    elseif i==n
        Sp(i)=3*S(i)/4;
    else
        Sp(i)=S(i)/2;
    end

    %Place collocation points at center of each panel except for
    %   panels near TE
```

```
    if i==1
        x(i)=X(i)+1/4*(X(i+1)-X(i));
        z(i)=Z(i)+1/4*(Z(i+1)-Z(i));
    elseif i==n
        x(i)=X(i)+3/4*(X(i+1)-X(i));
        z(i)=Z(i)+3/4*(Z(i+1)-Z(i));
    else
        x(i)=0.5*(X(i+1)+X(i));
        z(i)=0.5*(Z(i+1)+Z(i));
    end
end
rc=[x z];           %vectors from center of rotation to collocation
                    %   points

%Initialize and calculate influence variables
a=zeros(n); b=zeros(n); c=zeros(n); d=zeros(n); g=zeros(n);
h=zeros(n);
for i=1:n       %panel loop: rows of A matrix
    for j=1:n   %induction loop: columns of A matrix
        if j==i     %collocation point is on panel
            pan=1;
        else        %collocation point is not on panel
            pan=0;
        end
        [a(i,j),b(i,j),c(i,j),d(i,j),g(i,j),h(i,j)]=vor2dlc_u ...
        ([x(i),z(i)],[X(j),X(j+1)],[Z(j),Z(j+1)],al(j),S(j),pan);
    end
end

%Initialize and calculate influence coefficients to create Avorpan
%   matrix
Avorpan=zeros(n+1);
for i=1:n       %panel loop: rows of A matrix
    for j=1:n+1 %induction loop: columns of A matrix
        if j==1         %first entry of the row -> paired with
                        %   gamma(1)
            Avorpan(i,j)=a(i,1)*norm(i,1)+c(i,1)*norm(i,2);
        elseif j==n+1   %last entry of the row -> paired with
                        %   gamma(n+1)
            Avorpan(i,j)=b(i,n)*norm(i,1)+d(i,n)*norm(i,2);
        else            %other row entries -> paired with
                        %   gamma(2)-gamma(n)
            Avorpan(i,j)=(a(i,j)+b(i,j-1))*norm(i,1)+(c(i,j) ...
            +d(i,j-1))*norm(i,2);
        end
```

```
    end
end
%Kutta condition: gamma(1)+gamma(n+1)=0
Avorpan(n+1,1)=1;
Avorpan(n+1,n+1)=1;

%Write outputs
geometry.n=n;
geometry.X=X;
geometry.Z=Z;
geometry.TE=TE;
geometry.x=x;
geometry.z=z;
geometry.chord=chord;
geometry.qchord=qchord;
geometry.al=al;
geometry.norm=norm;
geometry.rc=rc;
geometry.S=S;
geometry.Sp=Sp;
coef.a=a;
coef.b=b;
coef.c=c;
coef.d=d;
coef.g=g;
coef.h=h;
coef.Avorpan=Avorpan;
```

B.2.2 Position, orientation, and velocity update

```
function [newloc,newvel,newgeom,newcoef]= ...
update_position(func,geometry,coef,loc0,vel0,t0,delt)
%function to update the position and orientation based on its
%   previous position and orientation, speed, and time step
%Input 'func' is a function that returns [U W thetadot] with time as
%   an input
%Input 'geometry' a data structure containing information about the
%   un-transformed geometry, taken from the output of 'init_comp.'
%   See that function's documentation for more details.
%Input 'coef' is a data structure containing coef.a, coef.b, coef.c,
%   and coef.d, taken from the output of 'init_comp.'  See that
%   function's documentation for more details.
%Input 'loc0' is a vector containing the instantaneous position and
```

```
%   orientation of the airfoil's coordinate axes with respect to the
%   global coordinate system.  The form is [X0 Z0 theta0].
%Input 'vel0' is a vector containing the linear and rotational
%   velocities of the airfoil's coordinate axes at the previous time
%   step, having the form [U0 W0 thetadot0].
%Input 't0' is the time at the previous time step
%Input 'delt' is the time step size.
%Output 'newloc' is the updated position and orientation of the
%   airfoil's coordinate axes.
%Output 'newvel' is the updated linear and rotational velocity of the
%   airfoil's coordinate axes.
%Output 'newgeom' is a data structure containing the same fields as
%   'geometry,' but transformed to align with the current position
%   and orientation of the airfoil's coordinate axes, 'newloc.'
%'Output 'newcoef' is a data structure containing the same fields as
%   'coef,' but transformed to align with the current position and
%   orientation of the airfoil's coordinate axes, 'newloc.'

%read geometry data
X=geometry.X;                   %panel endpoints (x)
Z=geometry.Z;                   %panel endpoints (z)
TE=geometry.TE;                 %TE location
x=geometry.x;                   %collocation points (x)
z=geometry.z;                   %collocation points (z)
n=size(x,1);                    %number of body panels
qchord=geometry.qchord;         %quarter-chord location
al=geometry.al;                 %panel angles

%read coef data
a=coef.a;
b=coef.b;
c=coef.c;
d=coef.d;

%Calculate new velocity
newvel=func(t0+delt);

%Update position and orientation of airfoil's coordinate axes
if t0<0
    newloc=loc0+vel0*delt;      %this case sets the initial
                                %   position and orientation of the
                                %   airfoil
else
    newloc=loc0+1/2*(vel0+newvel)*delt;
end
```

```
%Initialize new variables where required
Xnew=zeros(n+1,1);
Znew=zeros(n+1,1);
xnew=zeros(n,1);
znew=zeros(n,1);
normnew=zeros(n,2);        %panel normal directions
anew=zeros(n);
bnew=zeros(n);
cnew=zeros(n);
dnew=zeros(n);

%Transform geometry and coefficients to align with current position
%   of coordinate axes
for i=1:n
    Xnew(i)=newloc(1)+X(i)*cos(newloc(3))+Z(i)*sin(newloc(3));
    Znew(i)=newloc(2)-X(i)*sin(newloc(3))+Z(i)*cos(newloc(3));
    xnew(i)=newloc(1)+x(i)*cos(newloc(3))+z(i)*sin(newloc(3));
    znew(i)=newloc(2)-x(i)*sin(newloc(3))+z(i)*cos(newloc(3));
    normnew(i,:)=[sin(al(i)+newloc(3)) cos(al(i)+newloc(3))];
    for j=1:n
        anew(i,j)=a(i,j)*cos(newloc(3))+c(i,j)*sin(newloc(3));
        bnew(i,j)=b(i,j)*cos(newloc(3))+d(i,j)*sin(newloc(3));
        cnew(i,j)=-a(i,j)*sin(newloc(3))+c(i,j)*cos(newloc(3));
        dnew(i,j)=-b(i,j)*sin(newloc(3))+d(i,j)*cos(newloc(3));
    end
end
Xnew(n+1)=newloc(1)+X(n+1)*cos(newloc(3))+Z(n+1)*sin(newloc(3));
Znew(n+1)=newloc(2)-X(n+1)*sin(newloc(3))+Z(n+1)*cos(newloc(3));
qchordnew(1)=newloc(1)+qchord(1)*cos(newloc(3)) ...
+qchord(2)*sin(newloc(3));
qchordnew(2)=newloc(2)-qchord(1)*sin(newloc(3)) ...
+qchord(2)*cos(newloc(3));
TEnew(1)=newloc(1)+TE(1)*cos(newloc(3))+TE(2)*sin(newloc(3));
TEnew(2)=newloc(2)-TE(1)*sin(newloc(3))+TE(2)*cos(newloc(3));
alnew=al+ones(n,1)*newloc(3);
rcnew(:,1)=xnew-ones(n,1)*newloc(1);
rcnew(:,2)=znew-ones(n,1)*newloc(2);

%Write outputs
newgeom.X=Xnew;
newgeom.Z=Znew;
newgeom.TE=TEnew;
newgeom.x=xnew;
newgeom.z=znew;
```

```
newgeom.qchord=qchordnew;
newgeom.al=alnew;
newgeom.norm=normnew;
newgeom.rc=rcnew;
newcoef.a=anew;
newcoef.b=bnew;
newcoef.c=cnew;
newcoef.d=dnew;
```

B.2.3 Calculation of A, B, and C matrices

```
function [A,B,C,wakenew,foilcirc]=matrices ...
(geometry,vel,Avorpan,wake,prev_foilcirc)
%Function to compute A, B, and C matrices for solution of the system
%   of equations in the unsteady panel method.  A is an n+2 x n+2
%   matrix of influence coefficients of the unknown vortex strengths.
%   B is the vector of unknown vortex strengths.  C is the vector of
%   known normal velocities acting on each panel, plus the right-hand
%   sides of the Kutta condition and Kelvin's theorem.
%Output 'wakenew' is a data structure consisting of updates to
%   information about the wake, including the following fields:
    %wakenew.present: always equal to 1.  See note on input 'wake.'
    %wakenew.Gamw_new is the new vector of wake circulation strengths
    %   (including the k-th vortex)
    %wakenew.Gamw_sum_new is the amount of circulation in the wake
    %   for the current time step
    %wakenew.coef.e and wake.coef.f are influence coefficients of
    %   discrete vortex panels on collocation points, in the x- and z-
    %   directions
    %wakenew.coef.phiw are coefficients of velocity potential acting
    %   on the airfoil's collocation points
%Output 'foilcirc' is the bound airfoil circulation, which is required
%   for application of Kelvin's theorem during the subsequent time step
%Input 'geometry' is a data structure containing information about the
%   airfoil's position and orientation at the current time step - this
%   is 'geomnew' from the function 'update_position'
    %geometry.x: location of collocation points in global X coordinates
    %geometry.z: location of collocation points in global Z coordinates
    %geometry.norm: surface normal vectors
    %geometry.S: panel lengths
    %geometry.rc: relative vectors from center of rotation to vortex
    %   panels
%Input 'vel' is a vector containing the instantaneous velocity and
```

```
%   angular speed of the airfoil's coordinate system, [U W thetadot]
%Input 'Avorpan' is an n x n matrix of influence coefficients of the
%   surface vortex panels, computed in 'init_comp'
%Input 'wake' is a data structure containing the following information
%   about the wake:
    %wake.present: =1 if previous wake points exist, =0 otherwise
        %if wake.present==0, the calculation will be equivalent to a
        %   steady-state calculation and the output 'wakenew.present'
        %   will be set to 1
    %wake.xw: locations of wake vortices in global X coordinates
    %wake.zw: locations of wake vortices in global Z coordinates
    %wake.Gamw: wake circulation strengths for the first k-1 vortices
%Input 'prev_foilcirc' is the bound airfoil circulation from the
%   previous time step, required for application of Kelvin's theorem.
%   If t==0, the value of this input is not used and may be set to 0.

%read inputs
x=geometry.x;
z=geometry.z;
norm=geometry.norm;
S=geometry.S;
rc=geometry.rc;
n=size(S,1);

Cwake=zeros(n,1);
if wake.present==1
    xw=wake.xw;
    zw=wake.zw;
    Gamw=[wake.Gamw;0];
    k=size(xw,1);
    %initialize e and f coefficients and known influence sums
    e=zeros(n,k);
    f=zeros(n,k);
    theta=zeros(n,k);

    %Calculate e and f terms for wake contribution
    %Also calculate contribution of known wake terms on panels
    for i=1:n   %n is the number of airfoil panels
        for l=1:k   %k is the number of wake vortices
            [e(i,l),f(i,l),theta(i,l)]=vor2ddc([x(i),z(i)], ...
            [xw(l),zw(l)]);
            if l<k
                Cwake(i)=Cwake(i) ...
                +(e(i,l)*norm(i,1)+f(i,l)*norm(i,2))*Gamw(l);
            end
```

```
        end
    end
end

%Construct A matrix and C vector (A contains coefficients of unknown
%   quantities, C contains known quantities)
if wake.present==1
    A=ones(n+2,n+2);
    C=zeros(n+2,1);
    A(1:n+1,1:n+1)=Avorpan;     %Influence from vortex panels remains
                                %   constant and is calculated
                                %   previously
else
    A=Avorpan;
    C=zeros(n+1,1);
end
for i=1:n+1     %Compute last row and column of A matrix (Kelvin's
                %   theorem and last wake vortex influence,
                %   respectively)
    if i==1
        if wake.present==1
            A(n+2,i)=S(i)/2;
            A(i,n+2)=e(i,k)*norm(i,1)+f(i,k)*norm(i,2);
        end
        C(i)=(vel(1)+vel(3)*rc(i,2))*norm(i,1)+(vel(2) ...
        -vel(3)*rc(i,1))*norm(i,2)-Cwake(i);
    elseif i==n+1
        if wake.present==1
            A(n+2,i)=S(i-1)/2;
            A(i,n+2)=0;
        end
        C(i)=0;
    else
        if wake.present==1
            A(n+2,i)=0.5*(S(i-1)+S(i));
            A(i,n+2)=e(i,k)*norm(i,1)+f(i,k)*norm(i,2);
        end
        C(i)=(vel(1)+vel(3)*rc(i,2))*norm(i,1)+(vel(2) ...
        -vel(3)*rc(i,1))*norm(i,2)-Cwake(i);
    end
end
if wake.present==1
    C(n+2)=prev_foilcirc;
end
```

```
%Calculate B vector (contains unknown vortex panel strengths and
%   wake vortex strengths
B=A\C;

%calculate bound airfoil circulation for this time step
foilcirc=0;
for i=1:n
    foilcirc=foilcirc+0.5*(B(i)+B(i+1))*S(i);
end

if wake.present==1
%Apply last vortex strength and increase sum of wake vortex
%   circulation
    Gamw(k)=B(n+2);
    Gamw_new=Gamw;
end

%write outputs
wakenew.present=1;
if wake.present==1
    wakenew.Gamw=Gamw_new;
    wakenew.coef.e=e;
    wakenew.coef.f=f;
    wakenew.coef.theta=theta;
end
```

B.2.4 Computation of velocities, pressures, forces, moments

```
function [press,gradphi,dphi,forces]=press_comp ...
(stepnum,coef,gam,Gamw,upstream,vel,geometry,delt)
%Function to compute velocities and pressures at collocation points
%Input 'stepnum' is the iteration count.  If stepnum=0, no
%   information for coef.e, coef.f, coef.phiw, Gamw, or phi_old is
%   available.  Therefore, these fields are ignored for stepnum=0
%   and dphi/dt is assumed to be 0.
%Input 'coef' is a data structure containing the following fields:
    %coef.curr: coefficients for current time step
    %coef.old: coefficients from previous time step (if stepnum>0)
    %.a: see 'init_comp' output coef.a.  Requires transformed
    %   version from 'update_position'
    %.b: see 'init_comp' output coef.b.  Requires transformed
    %   version from 'update_position'
    %.c: see 'init_comp' output coef.c.  Requires transformed
```

```
    %   version from 'update_position'
    %.d: see 'init_comp' output coef.d.  Requires transformed
    %   version from 'update_position'
    %.e: see 'matrices' output wake.coef.e
    %.f: see 'matrices' output wake.coef.f
    %.g: see 'init_comp' output coef.g
    %.h: see 'init_comp' output coef.h
    %.theta: see 'matrices' output wake.coef.theta
%Input 'gam' is a nx2 vector of vortex panel endpoint strengths.
%   Requires data from this time step and previous one, if
%   stepnum>0.
%Input 'Gamw' is a kx2 vector of wake vortex strengths (current and
%   previous time steps)
%Input 'upstream' is comprised of [Pinf rho] from upstream
%   conditions
%Input 'vel' contains the current velocity and rotational velocity
%   from 'kinematics' in the form [U W thetadot]
%Input 'geometry' is a data structure containing the following
%   fields:
    %geometry.X: x-coordinates of airfoil vertices
    %geometry.Z: z-coordinates of airfoil vertices
    %geometry.x: x-coordinates of collocation points
    %geometry.z: z-coordinates of collocation points
    %geometry.S: airfoil panel lengths
    %geometry.al: panel angles
    %geometry.norm: airfoil surface normal vectors
    %geometry.chord: chord length of the airfoil
    %geometry.qchord: quarter-chord location for the airfoil
    %geometry.rc: vectors from the airfoil's center of rotation to
    %   collocation points
%Input 'delt' is the time step size
%Output 'press' is an nx2 matrix of pressures at the collocation
%   points, where row 1 is the absolute pressure and row 2 is the
%   gage pressure
%Output 'gradphi' is a matrix comprised of [u w] at the collocation
%   points
%Output 'phi' a vector containing the velocity potential at each
%   collocation point for the current time step
%Output 'forces' is a 1 x 3 array consisting of [L Dp M]:
    %L: total lift (vertical force) per unit span
    %Dp: total pressure drag (horizontal force) per unit span
    %M: total pitching moment about the quarter chord per unit span
%Note that lift and drag are referenced to the global vertical and
%   horizontal axes, respectively, since the direction of the relative
%   wind may change from one time step to the next.
```

```
%Read inputs
a=coef.curr.a;
b=coef.curr.b;
c=coef.curr.c;
d=coef.curr.d;
g1=coef.curr.g;
h1=coef.curr.h;
Pinf=upstream(1);
rho=upstream(2);
n=size(gam,1)-1;
X=geometry.X;
Z=geometry.Z;
qchord=geometry.qchord;
al=geometry.al;
norm=geometry.norm;
rc=geometry.rc;
S=geometry.S;
Sp=geometry.Sp;
if stepnum>0
    g2=coef.old.g;
    h2=coef.old.h;
    e=coef.curr.e;
    f=coef.curr.f;
    theta1=coef.curr.theta;
    if stepnum>1
        theta2=coef.old.theta;
    end
    k=size(Gamw,1);
end

%Compute u, v, velocity potential, time derivative of velocity
%   potential, and pressure at each collocation point
u=zeros(n,1);
w=zeros(n,1);
dphi=zeros(n,1);
press=zeros(n,3);
pressj=zeros(n+1,1);
for i=1:n
    for j=1:n   %influence of vortex panels
        u(i)=u(i)+a(i,j)*gam(j,1)+b(i,j)*gam(j+1,1);
        w(i)=w(i)+c(i,j)*gam(j,1)+d(i,j)*gam(j+1,1);
        if stepnum>0
            dphi(i)=dphi(i)+(g1(i,j)*gam(j,1)+h1(i,j)*gam(j+1,1) ...
            -g2(i,j)*gam(j,2)-h2(i,j)*gam(j+1,2));
```

```
        end
    end
    if stepnum>0
        for l=1:k-1     %influence of wake vortices
            u(i)=u(i)+e(i,l)*Gamw(l,1);
            w(i)=w(i)+f(i,l)*Gamw(l,1);
            %account for atan idiosyncrasies
            if quadrant(theta1(i,l))==4&&quadrant(theta2(i,l))==1
                dphi(i)=dphi(i)+(-Gamw(l,1)/(2*pi) ...
                *(2*pi-theta1(i,l))+Gamw(l,2)/(2*pi)*theta2(i,l));
            elseif quadrant(theta1(i,l))==1 ...
            &&quadrant(theta2(i,l))==4
                dphi(i)=dphi(i)+(-Gamw(l,1)/(2*pi)*theta1(i,l) ...
                +Gamw(l,2)/(2*pi)*(2*pi-theta2(i,l)));
            else
                dphi(i)=dphi(i)+(-Gamw(l,1)/(2*pi)*theta1(i,l) ...
                +Gamw(l,2)/(2*pi)*theta2(i,l));
            end
        end
        u(i)=u(i)+e(i,k)*Gamw(k,1);
        w(i)=w(i)+f(i,k)*Gamw(k,1);
        dphi(i)=dphi(i)-Gamw(k,1)/(2*pi)*theta1(i,k);

    else
        dphi(i)=0;
    end

    %Calculate pressure from unsteady Bernoulli equation
    press(i,1)=Pinf-rho*((u(i)^2+w(i)^2)/2-u(i)*(vel(1)+vel(3) ...
    *rc(i,2))-w(i)*(vel(2)-vel(3)*rc(i,1))+dphi(i)/delt);
    press(i,2)=press(i,1)-Pinf;     %gage pressure

    %Interpolate pressures to panel endpoints
    if i>1
        pressj(i)=press(i-1,1)+(press(i,1)-press(i-1,1)) ...
        /(Sp(i)+(S(i-1)-Sp(i-1)))*(S(i-1)-Sp(i-1));
    end
end
%Extrapolate pressures to trailing edge vertices
pressj(1)=press(1,1)-(pressj(2)-press(1,1))/(S(1)-Sp(1))*Sp(1);
pressj(n+1)=pressj(n)+(press(n,1)-pressj(n))/Sp(n)*S(n);

%Compute lift, drag, moment from pressure distribution
rf=zeros(n,2);          %vectors from quarter chord to panel centers
                        %   of pressure
```

```
Sf=zeros(n,1);          %lengths to centers of pressure from panel
                        %   endpoints
L=0;
Dp=0;
M=0;
for i=1:n
    Sf(i)=S(i)*(pressj(i)+2*pressj(i+1)) ...
    /(3*(pressj(i)+pressj(i+1)));               %Panel center of
                                                %   pressure (panel
                                                %   coordinates)
    rf(i,:)=[X(i)+Sf(i)*cos(al(i))-qchord(1), ...
     Z(i)-Sf(i)*sin(al(i))-qchord(2)];          %vector from quarter
                                                %   chord to panel
                                                %   center of
                                                %   pressure
    %panel lift from integration of surface pressure
    L=L-(pressj(i)+pressj(i+1))*(norm(i,1)*0+norm(i,2)*1)*S(i)/2;
    %panel drag from integration of surface pressure
    Dp=Dp-(pressj(i)+pressj(i+1))*(norm(i,1)*1+norm(i,2)*0)*S(i)/2;
    %panel contribution to moment about quarter-chord
    M=M+(pressj(i)+pressj(i+1))*S(i)/2*(rf(i,1)*norm(i,2) ...
    -rf(i,2)*norm(i,1));
end

%Output flow speed components at each collocation point
gradphi=[u w];

%Output forces and moments
forces=[L Dp M];
```

B.2.5 Computation of wake roll-up

```
function newwake=wake_rollup(geometry,gam,wake,delt)
%Function to compute the wake roll-up due to induced velocities
%   acting on discrete wake vortices
%Input 'geometry' is a data structure containing the following
%   fields:
    %geometry.X: global x-coordinates of airfoil vertices
    %geometry.Z: global z-coordinates of airfoil vertices
    %geometry.al: panel angles in global coordinates
    %geometry.S: panel lengths
%Input 'gam' is a vector of strengths of the endpoints of the vortex
%   panels lining the airfoil surface
```

```
%Input 'wake' is a data structure containing the following fields:
    %wake.xw: locations of wake vortices, in global x-coordinates
    %wake.zw: locations of wake vortices, in global z-coordinates
    %wake.Gamw: vector of wake vortex strengths
%Input 'delt' is the time step size
%Output 'newwake' is a data structure containing updated values of
%   wake.xw and wake.zw based on wake roll-up computations

%read inputs
X=geometry.X;
Z=geometry.Z;
al=geometry.al;
S=geometry.S;
xw=wake.xw;
zw=wake.zw;
Gamw=wake.Gamw;
n=size(S,1);
k=size(Gamw,1);

%Compute induced velocity at each wake vortex due to airfoil
%   circulation distribution and other wake vortices.  Move wake
%   vortices according to Pnew(wake) = Pold(wake) +
%   Vinduced(wake)*delt
uw=zeros(k,1);ww=zeros(k,1);
aw=zeros(n,k);bw=zeros(n,k);cw=zeros(n,k);dw=zeros(n,k);
ew=zeros(k,k);fw=zeros(k,k);
xwnew=zeros(k,1);zwnew=zeros(k,1);
for l=1:k   %Wake vortex loop - influence on
    for i=1:n   %Airfoil panel loop - influence from
        [aw(i,l),bw(i,l),cw(i,l),dw(i,l),~,~]=vor2dlc_u ...
        ([xw(l) zw(l)],[X(i) X(i+1)],[Z(i) Z(i+1)],al(i),S(i),0);
        uw(l)=uw(l)+aw(i,l)*gam(i)+bw(i,l)*gam(i+1);
        ww(l)=ww(l)+cw(i,l)*gam(i)+dw(i,l)*gam(i+1);
    end
    for j=1:k   %Wake vortex loop - influence from
        if j==l     %Vortex has no influence on itself
            ew(j,l)=0;
            fw(j,l)=0;
        else
            [ew(j,l),fw(j,l),~]=vor2ddc([xw(l) zw(l)],[xw(j) zw(j)]);
        end
        uw(l)=uw(l)+ew(j,l)*Gamw(j);
        ww(l)=ww(l)+fw(j,l)*Gamw(j);
    end
    %Compute new location for the l-th wake vortex
```

```
    xwnew(l)=xw(l)+uw(l)*delt;
    zwnew(l)=zw(l)+ww(l)*delt;

    %error checking
    if isnan(xwnew(l))==1||isnan(zwnew(l))==1
        error(['Two consecutive wake points are directly on top of' ...
        ' each other, creating a singularity.  This occurs under' ...
        ' certain conditions when the trailing edge does not move' ...
        ' during a time step.  In most cases this can be fixed by' ...
        ' adjusting the velocity profile slightly to remove this' ...
        ' numerical issue.'])
    end
end

%Write outputs
newwake.xw=xwnew;
newwake.zw=zwnew;
```

B.3 Third-level functions

B.3.1 Computation of vortex-panel influences

```
function [a,b,c,d,g,h]=vor2dlc_u(P,X,Z,al,S,pan)
%function to calculate factors a, b, c, d, g, h for a point P due to
%   a single linear-strength vortex panel with endpoints (X(1), Z(1))
%   and (X(2), Z(2)), panel angle al, and panel length S
%'pan' is either 1 or 0: 1 if the point P is on the panel, 0 if not
%a - d are coefficients for induced velocity calculations; g and h
%   are coefficients to compute stream function.
%intended for us with the unsteady panel code

%Transform collocation point to panel coordinates
Pp=[cos(al) -sin(al);sin(al) cos(al)]*[P(1)-X(1);P(2)-Z(1)];

%Calculate distances r from panel edges and angles th for
%   collocation point
r=[sqrt((P(1)-X(1))^2+(P(2)-Z(1))^2), ...
sqrt((P(1)-X(2))^2+(P(2)-Z(2))^2)];
th=[atan2(Pp(2),Pp(1)) atan2(Pp(2),Pp(1)-S)];

%Determine whether the point P is on the vortex panel
%Calculate a, b, c, d, g, h in panel coordinates
```

```
if pan==1  %on vortex panel
    ap=(1-Pp(1)/S)/2;
    bp=Pp(1)/(2*S);
    g=0.5*(Pp(1)*(1-Pp(1)/(2*S))-S/2);
    h=0.25/S*(Pp(1)^2-S^2);
else
    ap=-1/(2*pi*S)*(Pp(2)*log(r(2)/r(1))+(Pp(1)-S)*(th(2)-th(1)));
    bp=1/(2*pi*S)*(Pp(2)*log(r(2)/r(1))+Pp(1)*(th(2)-th(1)));
    g=-1/(2*pi)*((Pp(1)*th(1) ...
    -(Pp(1)-S)*th(2)+Pp(2)*log(r(1)/r(2))) ...
    -1/S*(Pp(1)*Pp(2)*log(r(1)/r(2))-S*Pp(2)/2 ...
    +(Pp(1)^2-Pp(2)^2)/2*th(1)-(Pp(1)^2-Pp(2)^2-S^2)/2*th(2)));
    h=-1/(2*pi*S)*(Pp(1)*Pp(2)*log(r(1)/r(2)) ...
    -S*Pp(2)/2+(Pp(1)^2-Pp(2)^2)/2*th(1) ...
    -(Pp(1)^2-Pp(2)^2-S^2)/2*th(2));
end
cp=1/(2*pi*S)*((S-Pp(1))*log(r(2)/r(1))-S+Pp(2)*(th(2)-th(1)));
dp=1/(2*pi*S)*(Pp(1)*log(r(2)/r(1))+S-Pp(2)*(th(2)-th(1)));

%Transform a, b, c, d to global coordinates
vect=[cos(al) 0 sin(al) 0
    0 cos(al) 0 sin(al)
    -sin(al) 0 cos(al) 0
    0 -sin(al) 0 cos(al)]*[ap;bp;cp;dp];
a=vect(1); b=vect(2); c=vect(3); d=vect(4);
```

B.3.2 Computation of discrete-vortex influences

```
function [e,f,th]=vor2ddc(P,Pw)
%function to calculate the coefficients e and f due to the (influence
%   of a discrete vortex located at point Pw acting on a point P.  The
%   velocity induced at point P is equal to e*Gamma in the x-direction
%   and f*Gamma in the z-direction, where Gamma is the strength of the
%   discrete vortex
%th is the angle between the vortex and the collocation point, used
%   for calculating the wake contribution to velocity potential.  Goes
%   from 0 to 2*pi.

e=(P(2)-Pw(2))/(2*pi*((P(1)-Pw(1))^2+(P(2)-Pw(2))^2));
f=-(P(1)-Pw(1))/(2*pi*((P(1)-Pw(1))^2+(P(2)-Pw(2))^2));

th=atan2(P(2)-Pw(2),P(1)-Pw(1));
if th<0
```

```
    th=2*pi+th;
end
```

B.3.3 Quadrant computation for tan^{-1}

```
function number=quadrant(angle)
%Returns the quadrant of an angle (in radians)

if 0<=angle&&angle<pi/2
    number=1;
elseif pi/2<=angle&&angle<pi
    number=2;
elseif pi<=angle&&angle<3*pi/2
    number=3;
else
    number=4;
end
```

B.4 Example test case: accelerating airfoil

B.4.1 Kinematics function

```
function vel=kinematics_accel(t)

U=-30*t;          %horizontal velocity (to the left)
W=0;              %vertical velocity
thetadot=0;       %clockwise rotational velocity

vel=[U W thetadot];
```

B.4.2 Workspace inputs

```
%NACA 0006 geometry with 80 total points
%   Note trailing edge thickness must be 0
naca0006_80=[1.0000         0
    0.9871    0.0009
    0.9602    0.0028
```

```
 0.9323    0.0046
 0.9044    0.0064
 0.8765    0.0081
 0.8486    0.0098
 0.8207    0.0114
 0.7928    0.0130
 0.7649    0.0145
 0.7370    0.0160
 0.7091    0.0174
 0.6812    0.0188
 0.6532    0.0201
 0.6253    0.0214
 0.5974    0.0226
 0.5695    0.0237
 0.5416    0.0247
 0.5137    0.0257
 0.4858    0.0266
 0.4579    0.0274
 0.4300    0.0281
 0.4021    0.0287
 0.3743    0.0292
 0.3464    0.0296
 0.3186    0.0298
 0.2908    0.0298
 0.2630    0.0297
 0.2352    0.0293
 0.2075    0.0288
 0.1799    0.0279
 0.1523    0.0267
 0.1248    0.0252
 0.0976    0.0232
 0.0707    0.0205
 0.0446    0.0169
 0.0218    0.0123
 0.0087    0.0080
 0.0028    0.0046
 0.0003    0.0015
 0.0003   −0.0015
 0.0028   −0.0046
 0.0087   −0.0080
 0.0218   −0.0123
 0.0446   −0.0169
 0.0707   −0.0205
 0.0976   −0.0232
 0.1248   −0.0252
```

```
    0.1523   -0.0267
    0.1799   -0.0279
    0.2075   -0.0288
    0.2352   -0.0293
    0.2630   -0.0297
    0.2908   -0.0298
    0.3186   -0.0298
    0.3464   -0.0296
    0.3743   -0.0292
    0.4021   -0.0287
    0.4300   -0.0281
    0.4579   -0.0274
    0.4858   -0.0266
    0.5137   -0.0257
    0.5416   -0.0247
    0.5695   -0.0237
    0.5974   -0.0226
    0.6253   -0.0214
    0.6532   -0.0201
    0.6812   -0.0188
    0.7091   -0.0174
    0.7370   -0.0160
    0.7649   -0.0145
    0.7928   -0.0130
    0.8207   -0.0114
    0.8486   -0.0098
    0.8765   -0.0081
    0.9044   -0.0064
    0.9323   -0.0046
    0.9602   -0.0028
    0.9871   -0.0009
    1.0000         0];

%Other user-supplied inputs
body=flipud(naca0006_80);     %airfoil geometry
body(:,1)=body(:,1)-0.25;     %shift geometry relative to body-axis
upstream=[1.01325e5 1.225];   %upstream pressure and density
delt=0.01;                    %time step size
nsteps=90;                    %number of time steps
pitch_init=10;                %initial pitch angle above -X axis

%command to run the simulation
[results,forces,time,CPUtime]= ...
ULVPC(body,upstream,pitch_init,@kinematics_accel,delt,nsteps);
```

B.4.3 Creating a movie of the simulation

```
X=results(nsteps+1).geometry.X;
Z=results(nsteps+1).geometry.Z;
xw=results(nsteps+1).wake.xw;
zw=results(nsteps+1).wake.zw;
plot(X,Z,xw,zw,'x'),axis equal
v=axis;

for i=1:nsteps+1
    if i>1
        Gamw=results(i).wake.Gamw;
        k=size(Gamw,1);
        xw=results(i).wake.xw;
        zw=results(i).wake.zw;
        xwc=zeros(k,1);
        zwc=zeros(k,1);
        xwcc=zeros(k,1);
        zwcc=zeros(k,1);
    else
        k=0;
    end
    countc=1;
    countcc=1;
    cact=0;
    ccact=0;
    for l=1:k
        if Gamw(l)<=0
            xwcc(countcc)=xw(l);
            zwcc(countcc)=zw(l);
            ccact=1;
            countcc=countcc+1;
        else
            xwc(countc)=xw(l);
            zwc(countc)=zw(l);
            cact=1;
            countc=countc+1;
        end
    end
    X=results(i).geometry.X;
    Z=results(i).geometry.Z;
    if cact==0&&ccact==0        %No wake
        plot(X,Z,'b'),axis(v);
    elseif cact==1&&ccact==0    %Only clockwise vortices in wake
```

```
        plot(X,Z,'b',xwc(1:countc-1),zwc(1:countc-1),'xg'),axis(v)
    elseif cact==0&&ccact==1    %Only counterclockwise vortices in
                                %   wake
        plot(X,Z,'b',xwcc(1:countcc-1),zwcc(1:countcc-1),'xr'),axis(v)
    else                        %Counterclockwise and clockwise
                                %   vortices in wake
        plot(X,Z,'b',xwc(1:countc-1),zwc(1:countc-1),'xg', ...
        xwcc(1:countcc-1),zwcc(1:countcc-1),'xr'),axis(v)
    end
    F(i)=getframe;
end
```